НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

«На правах рукопису»
УДК 004.932

«До захисту допущено»
Науковий керівник кафедри
_____ І.А. Дичка
«___»_____2018 р.

Магістерська дисертація

на здобуття ступеня магістра

зі спеціальності 121 Інженерія програмного забезпечення

на тему: «Спосіб та програмне забезпечення для генерації освітлення та тіней об'єктів доповненої реальності»

Виконав:
студент VI курсу, групи КП-71мп
Дробот Олександр Олександрович _____

Керівник:
Доцент кафедри ПЗКС, к.т.н., доцент,
Сулема Є.С. _____

Рецензент:
к.т.н., доцент, в.о. завідувача
кафедри ММСА ІПСА
Тимощук О.Л. _____

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.
Студент _____

Київ – 2018 року

Національний технічний університет України

«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет прикладної математики

Кафедра програмного забезпечення комп'ютерних систем

Рівень вищої освіти – другий (магістерський) за освітньо-науковою програмою

Спеціальність (спеціалізація) – 121 «Інженерія програмного забезпечення» («Програмне забезпечення комп'ютерних та інформаційно-пошукових систем»)

ЗАТВЕРДЖУЮ

Науковий керівник кафедри

_____ І.А. Дичка

«___»_____ 2017 р.

ЗАВДАННЯ
на магістерську дисертацію студенту

Дроботу Олександру Олександровичу

1. Тема дисертації «Спосіб та програмне забезпечення для генерації освітлення та тіней об'єктів доповненої реальності», науковий керівник дисертації доцент кафедри ПЗКС, к.т.н., Сулема Євгенія Станіславівна затверджені наказом по університету від «___»_____ 2018 р. №_____

2. Термін подання студентом дисертації «14» грудня 2018 р.

3. Об'єкт дослідження: процес генерації освітлення та тіней об'єктів доповненої реальності.

4. Предмет дослідження: способи генерації освітлення та тіней доповненої реальності.

5. Перелік завдань, які потрібно розробити:
   - провести аналіз існуючих способів та алгоритмів генерації світла та тіней;
   - розробити модифікований спосіб генерації освітлення та тіней об'єктів доповненої реальності;
   - розробити архітектуру програмного забезпечення;
   - розробити програмне забезпечення, що реалізує запропонований спосіб генерації освітлення та тіней об'єктів доповненої реальності;
   - провести дослідження ефективності розробленого способу генерації освітлення та тіней об'єктів доповненої реальності.

6. Орієнтовний перелік публікацій:
   - XI наукова конференція магістрантів та аспірантів «Прикладна математика та комп'ютинг» (ПМК-2018-2).

7. Дата видачі завдання «15» грудня 2017 р.

Календарний план

№ з/п	Назва етапів виконання магістерської дисертації	Термін виконання етапів магістерської дисертації	Примітка
1.	Ґрунтовне ознайомлення з предметною галуззю	17.02.2018	
2.	Визначення структури магістерської дисертації; вивчення літератури, пошук додаткової літератури, патентний пошук	04.03.2018	
3.	Робота над першим розділом магістерської дисертації; проведення наукового дослідження	16.04.2018	
4.	Проведення наукового дослідження; робота над другим розділом магістерської дисертації; розроблення програмного забезпечення	14.05.2018	
5.	Проведення наукового дослідження; робота над статтею за результатами наукового дослідження	17.09.2018	
6.	Проведення наукового дослідження; робота над третім розділом магістерської дисертації; підготовка матеріалів доповіді на конференції ПМК-2018.	02.10.2018	
7.	Завершення роботи над основною частиною магістерської дисертації; підготовка ілюстративного матеріалу;	28.10.2018	
8.	Оформлення текстової і графічної частини магістерської дисертації	10.11.2018	

Студент О.О. Дробот

Науковий керівник Є.С Сулема

РЕФЕРАТ

Актуальність. Останнім часом спостерігається стрімкий розвиток технологій доповненої реальності. Згідно з прогнозом агенції Gartner ринок застосунків, що використовують елементи доповненої реальності, щорічно розширюватиметься, зокрема, за рахунок появи нових галузей застосування, в тому числі, в інженерії. В інженерних задачах доповнена реальність буде застосовуватись для покращення рівня безпеки та умов роботи з реальними технічними об'єктами, наприклад, шляхом накладання текстових інструкцій, графічних попереджень, додаткових зображень на реальні зображення технічного обладнання, яке спостерігає робітник через пристрої доповненої реальності. Тому якість відображення елементів доповненої реальності є актуальною задачею. Аналіз існуючих алгоритмів генерації освітлення доповненої реальності показав, що всі проаналізовані алгоритми мають свої недоліки, зокрема, вони не дозволяють генерувати освітлення на основі навколишнього середовища. Тому у даній роботі пропонується вдосконалений спосіб генерації освітлення та тіней об'єктів доповненої реальності, який вирішує проблеми, виявлені у проаналізованих алгоритмах.

Об'єктом дослідження в даній роботі є процес генерації освітлення та тіней об'єктів доповненої реальності.

Предметом дослідження є способи та засоби генерації освітлення та тіней об'єктів доповненої реальності.

Метою дослідження є аналіз існуючих способів генерації освітлення та тіней об'єктів доповненої реальності, їх властивостей, особливостей з подальшим створенням власного способу.

Методи дослідження. В роботі використовуються методи комп'ютерного моделювання, статистичні та емпіричні методи.

Наукова новизна роботи:

1. Запропоновано удосконалений спосіб генерації освітлення та тіней об'єктів доповненої реальності, який на відміну від існуючих способів використовує збір даних про рівень освітлення оточуючого

середовища для визначення домінуючого напряму світла, згідно з яким генерувати тінь об’єкту доповненої реальності.

2. Запропоновано модифікований алгоритм Light Estimation, який на відміну від базового алгоритму Light Estimation дозволяє обчислювати рівень освітлення для об’єктів доповненої реальності.

Практична цінність роботи полягає у збільшенні якості відображення об’єктів доповненої реальності.

Апробація роботи. Основні положення і результати роботи доповідалися та обговорювалися на X науковій конференції магістрантів та аспірантів «Прикладна математика та комп’ютинг» ПМК-2018.

Структура та обсяг роботи. Магістерська дисертація складається з вступу, п’яти розділів, висновків та додатків.

У вступі наведена охарактеризовано галузь використання та сферу застосування розробленого способу.

У першому розділі проведено аналіз існуючих програмних рішень для використання доповненої реальності, а також способи та алгоритми для генерації освітлення для об’єктів доповненої реальності.

У другому розділі описано створений спосіб генерації освітлення та тіней об’єктів доповненої реальності та описано алгоритм роботи розробленого способу.

У третьому розділі описано архітектуру програмного забезпечення, описано набір програмних засобів, які необхідні для створення способу.

У четвертому розділі були проаналізовані результати роботи способу, наведені приклади роботи способу, а також порівняння розробленого способу з існуючими.

У п’ятому розділі наведена побудова бізнес-моделі, що обґрунтовує доцільність реалізованого програмного забезпечення та прогнозує його потенційну прибутковість у майбутньому.

У висновках проаналізовано отримані результати роботи.

Робота виконана на 80 аркушах, містить посилання на список використаних літературних джерел.

Ключові слова: доповнена реальність, комп’ютерна графіка, обробка зображень.

ABSTRACT

Topicality. Recently, there has been a rapid development of technologies of augmented reality. According to Gartner's forecast, the application market, using elements of the augmented reality, will increase annually, due in particular to the emergence of new applications, including engineering. In engineering, the augmented reality will be used to improve the level of security and working conditions with real technical objects, for example, by imposing text instructions, graphic warnings, additional images on real hardware images that are observed by the worker through augmented reality devices. Therefore, the quality of display of elements of the augmented reality is an actual task. An analysis of the existing algorithms for generating illumination of augmented reality showed that all analyzed algorithms have their disadvantages, in particular, they do not allow the generation of lighting on the basis of the environment. Therefore, in this paper, proposed an improved way of generating illumination and shadows of objects of the augmented reality, which solves the problems found in the analyzed algorithms.

The object of research in this paper is the process of generating lighting and the shadows of objects of augmented reality.

The subject of the study is the ways and means of generating the illumination and shadows of the objects of the augmented reality.

The purpose of the study is to analyze the existing ways of generating lighting and shadows of the objects of the augmented reality, their properties, features, and the subsequent creation of their own method.

Research methods. Methods of computer modeling, statistical and empirical methods are used in this work.

The scientific novelty:

1. An improved way of generating illumination and shadows of the objects of augmented reality is proposed, which, unlike the existing methods, uses the collection of data on the level of ambient lighting to determine the dominant light direction and generates the shadow of the augmented reality object according to this dominant light.

2. The modified Light Estimation algorithm is proposed, which, unlike the basic Light Estimation algorithm, allows to calculate the level of illumination for objects of the augmented reality.

The practical value of the work is to increase the quality of the visualization of the objects of the augmented reality.

Test work. The main provisions and results of work were reported and discussed at the Xth International Conference of Masters and Postgraduates "Applied Mathematics and Computer", PMK-2018.

Structure and scope of work. The master's dissertation consists of an introduction, five sections, conclusions and appendices.

In the introduction, we describe the field of application and the scope of the developed method.

In the first section, an analysis of existing software solutions for the use of the augmented reality, as well as methods and algorithms for generation of lighting for objects of the complement of reality was conducted.

The second section describes the created way of generating the illumination and shadows of the objects of the complemented reality and describes the algorithm of the work of the developed method.

The third section describes the software architecture, describes the set of software tools that are required to create the method.

In the fourth section, the results of the method work were analyzed, examples of work of the method are given, as well as comparison of the developed method with the existing ones.

The fifth section presents the construction of a business model that justifies the feasibility of the software and predicts its potential profitability in the future.

The results of work are analyzed in the conclusions.

The work is done on 80 sheets, contains a link to the list of used literary sources.

Keywords: augmented reality, computer graphics, image processing.

РЕФЕРАТ

Актуальность. В последнее время наблюдается стремительное развитие технологий дополненной реальности. Согласно прогнозу агентства Gartner рынок приложений, использующих элементы дополненной реальности, ежегодно расширяться, в частности, за счет появления новых областей применения, в том числе, в инженерии. В инженерных задачах дополненная реальность будет применяться для улучшения уровня безопасности и условий работы с реальными техническими объектами, например, путем наложения текстовых инструкций, графических предупреждений, дополнительных изображений на реальные изображения технического оборудования, которое наблюдает рабочий через устройства дополненной реальности. Поэтому качество отображения элементов дополненной реальности является актуальной задачей. Анализ существующих алгоритмов генерации освещения дополненной реальности показал, что все проанализированные алгоритмы имеют свои недостатки, в частности, они не позволяют генерировать освещения на основе окружающей среды. Поэтому в данной работе предлагается усовершенствованный способ генерации освещения и теней объектов дополненной реальности, который решает проблемы, выявленные в проанализированных алгоритмах.

Объектом исследования в данной работе является процесс генерации освещения и теней объектов дополненной реальности.

Предметом исследования являются способы и средства генерации освещения и теней объектов дополненной реальности.

Целью исследования является анализ существующих способов генерации освещения и теней объектов дополненной реальности, их свойств, особенностей с последующим созданием собственного образа.

Методы исследования. В работе используются методы компьютерного моделирования, статистические и эмпирические методы.

Научная новизна работы:

1. Предложен усовершенствованный способ генерации освещения и теней объектов дополненной реальности, который в отличие от существующих способов использует сбор данных об уровне освещения окружающей среды для определения доминирующего направления света, согласно которому генерируется тень объекта дополненной реальности.
2. Предложен модифицированный алгоритм Light Estimation, который в отличие от базового алгоритма Light Estimation позволяет вычислять уровень освещения для объектов дополненной реальности.

Практическая ценность работы заключается в увеличении качества отображения объектов дополненной реальности.

Апробация работы. Основные положения и результаты работы докладывались и обсуждались на X научной конференции магистрантов и аспирантов «Прикладная математика и компьютинг» ПМК-2018.

Структура и объем работы. Магистерская диссертация состоит из введения, пяти глав, заключения и приложений.

Во введении приведена охарактеризованы область применения и область применения разработанного способа.

В первой главе проведен анализ существующих программных решений для использования дополненной реальности, а также способы и методы для генерации освещения для объектов дополненной реальности

Во втором разделе описано создан способ генерации освещения и теней объектов дополненной реальности и описан алгоритм работы разработанного способа.

В третьем разделе описано архитектуру программного обеспечения, описано набор программных средств, которые необходимы для создания образа.

В четвертом разделе были проанализированы результаты работы способа, приведены примеры работы способа, а также сравнение разработанного способа с существующими.

В пятом разделе приведена построение бизнес-модели, обосновывает целесообразность реализованного программного обеспечения прогнозирует его потенциальную прибыльность в будущем.

В выводах проанализированы полученные результаты работы.

Работа выполнена на 80 листах, содержит ссылки на список использованных литературных источников.

Ключевые слова: дополненная реальность, компьютерная графика, обработка изображений.

ЗМІСТ

СПИСОК ТЕРМІНІВ, СКОРОЧЕНЬ ТА ПОЗНАЧЕНЬ

AR – augmented reality або доповнена реальність.

VR – virtual reality або віртуальна реальність.

MR – mixed reality або змішана реальність.

IMU – inertial measurement unit або інерційний вимірювальний пристрій.

Якір – anchor або точка прив'язки позиції.

Хіт-тест – вектор позиції на площині доповненої реальності, який був отриманий у результаті дотику до екрану.

API – прикладний програмний інтерфейс.

SDK (від англ. Software Development Kit) –набір із засобів розроблення.

Програмний рушій (англ. software engine) або частіше просто рушій – у програмуванні – ядро комп'ютерної програми для реалізації конкретної прикладної задачі, щоб відрізнити її від наповнення і зовнішнього вигляду конкретної програми. Рушій може мати форму програми, частини програми, комплексу програм або бібліотеки, в залежності від завдання і реалізації. Як правило, рушій виділяється з програми для використання в декількох проектах або роздільної розроблення та тестування.

GUI – тип інтерфейсу, який дозволяє користувачам взаємодіяти з електронними пристроями через графічні зображення та візуальні вказівки, на відміну від текстових інтерфейсів, заснованих на використанні тексту, текстовому наборі команд та текстовій навігації.

SLAM – (англ. simultaneous localization and mapping — SLAM) одночасна локалізація і картографування.

COM – (англ. concurrent odometry and mapping) паралельна одометрія та відображення.

SceneKit – бібліотека для розроблення тривимірних програм під платформу iOS

Metal – це низькорівневий прискорювач 3D-графіки та обчислювальний інтерфейс прикладного програмування шейдерів, розроблений компанією Apple Inc., який дебютував у iOS 8. Метал поєднує функції, подібні до

OpenGL та OpenCL під одним API. Він створений для додатків розроблених під iOS, macOS та tvOS і є подібним до API на інших платформах, таких як Vulkan і DirectX 12.

Vulkan — багатоплатформна API для 3D графіки і супроводжуючих обчислень, представлене компанією Khronos Group.

ВСТУП

Доповнена реальність – це середовище з прямим або непрямим доповненням фізичного світу цифровими даними. Процес додавання об’єктів доповненої реальності відбувається в режимі реального часу, а провідником служать цифрові пристрої – планшети, смартфони, «розумні» окуляри або аксесуари зі спеціальним програмним забезпеченням. Пристрій через камеру або інший інтерфейс (наприклад, Bluetooth або GPS) розпізнає об’єкти, особливі мітки на них або місце розташування користувача. Маркером може бути як спеціальний радіомаячок з вбудованим чіпом, так і звичайний QR-код. Для того, щоб відбулося доповнення, користувачеві потрібно потрапити в зону дії цього маячка або зчитати QR-код.

Додатки на основі доповненої реальності можуть допомогати людині фокусувати увагу на певних елементах зображення, що отримується з камери; покращувати розуміння об’єктів оточуючого світу шляхом надавання необхідної інформації, що накладається на зображення у вигляді текстового повідомлення або візуального образу.

На сьогодні доповнена реальність використовується у таких сферах діяльності: освіта, медицина, військова справа, авіація, маркетинг, туризм, дизайн та ігри. У сфері освіти доповнена реальність допомагає дітям пізнавати світ, оскільки вони можуть навести камеру телефону на предмет зацікавленості і побачити детальну інформацію про нього. У військовій справі доповнена реальність допомагає у підготовці військовослужбовців шляхом навчання на спеціальних полігонах з використанням доповненої реальності. У авіації доповнена реальність допомагає навчати нових пілотів, а також слугує навігатором. У сфері маркетингу доповнена реальність допомагає продавати різноманітні товари та послуги шляхом показу цих товарів за допомогою спеціальної реклами. У сфері туризму доповнена реальність може допомагати користувачеві у пошуку шляху та знаходити і виводити інформацію про архітектурну будівлю або інший туристичний

об’єкт, на який користувач наводить свій мобільний телефон. У сфері дизайну за допомогою доповненої реальності можна, наприклад, зручно презентувати архітектуру розробленої будівлі будь-якого розміру.

Отже, доповнена реальність має широке коло застосувань. Проте, на сьогодні важливою є проблема якісного відображення об’єктів у доповненій реальності. Заокрема, важливою складовою у відтворенні доповненої реальності є коректна генерація освітлення та тіней, оскільки від цього залежить якість сприйняття доповненої реальності користувачем.

Існуючі програмні рішення даної проблеми на сьогодні не дозволяють отримати достатньо реалістичний результат візуалізації об’єктів доповненої реальності. Тому задача розроблення нових, вдосконалених способів генерації освітлення та тіней є актуальною.

В даній магістерській дисертації представлено результат дослідження з вдосконалення способу генерації освітлення та тіней елементів доповненої реальності.

РОЗДІЛ 1 АНАЛІЗ ІСНУЮЧИХ СПОСОБІВ ТА ПРОГРАМНИХ ЗАСОБІВ ДЛЯ РОБОТИ З ДОПОВНЕНОЮ РЕАЛЬНІСТЮ

1.1 Аналіз існуючих бібліотек та програм для роботи з доповненою реальністю

Розглянемо існуючі програмні засоби, які використовуються для взаємодії мобільних пристроїв із технологією доповненої реальності.

1.1.1 Google Arcore

ARCore – це платформа Google для створення взаємодії з доповненою реальністю [1]. Розглянемо її основні елементи.

Відстеження руху

Коли телефон переміщується по навколишньому середовищу, ARCore використовує процес, який називається паралельною одометрією та відображенням, або COM, щоб зрозуміти, де телефон знаходиться у просторі в поточний момент часу. ARCore виявляє візуально окремі частини зображень з камери, які називаються функціональними точками і використовує ці точки для обчислення зміни в розташуванні. Візуальна інформація поєднується з інерційними вимірами з IMU датчику для оцінки позиції (положення та орієнтації) камери відносно позиції у просторі. На рис. 1 можна побачити схематичні можливості IMU датчику.

Вирівнюючи позицію віртуальної камери, яка відтворює 3D-вміст на позицію камери пристрою, наданої ARCore, розробники можуть відображати віртуальний зміст з правильної точки зору. Відтворене віртуальне зображення може бути накладено поверх зображення, отриманого з камери пристрою, і воно виглядає так, ніби віртуальний вміст є частиною реального світу.

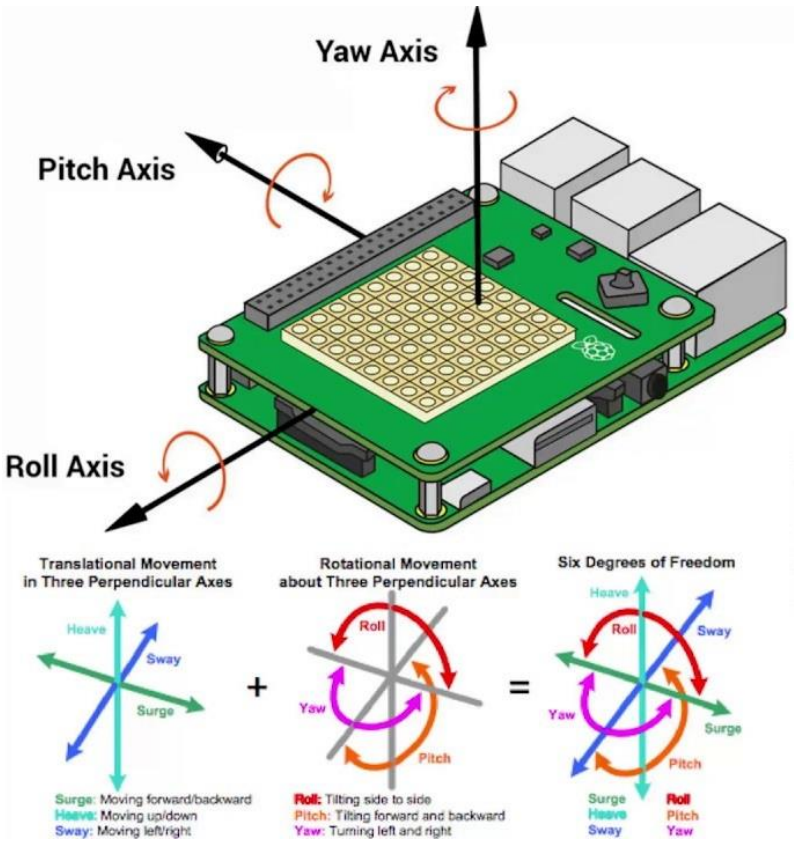


Рис. 1. Схематичні можливості IMU

Аналіз навколишнього середовища

ARCore шукає кластери характерних точок, які з'являються на горизонтальних і вертикальних поверхнях, таких як столи або стіни і робить ці поверхні доступними для застосування користувачем у якості площин віртуальної реальності (рис. 2). ARCore також може визначити межу кожної площини та зробити цю інформацію доступною для додатка користувача. Користувач може використовувати цю інформацію для розміщення віртуальних об'єктів на рівних поверхнях.

Оскільки ARCore використовує ознаки точок для аналізу площин, то такі плоскі поверхні без текстури як біла стіна, можуть бути не знайдені належним чином. ARCore може погано виявляти площини при низькому освітленні.

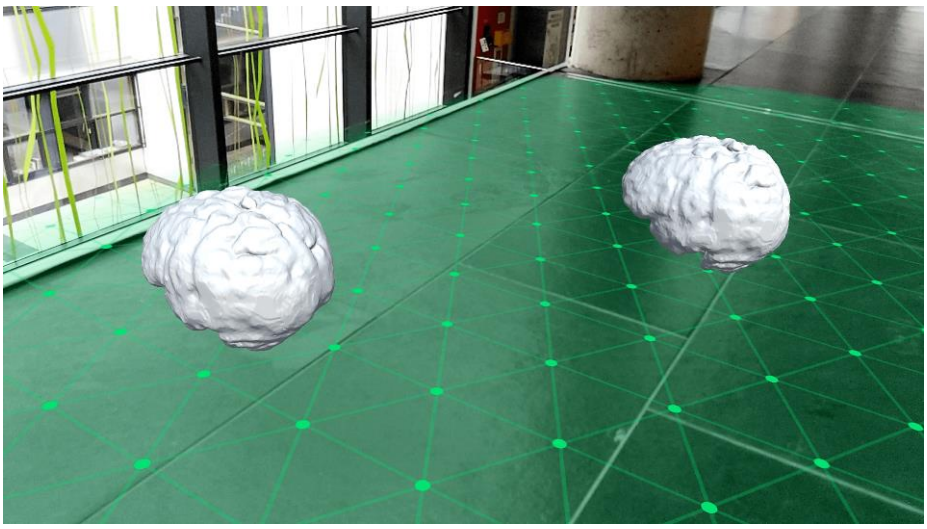


Рис. 2. Аналіз навколишнього середовища ARcore

Оцінка світла

ARCore може проаналізувати інформацію про освітлення оточення та забезпечити користувачеві середню інтенсивність та корекцію кольору даного зображення з камери (рис. 3). Ця інформація дозволяє користувачеві освітлювати свої віртуальні об'єкти за тих самих умов, що й оточення навколо них, підвищуючи відчуття реалізму.

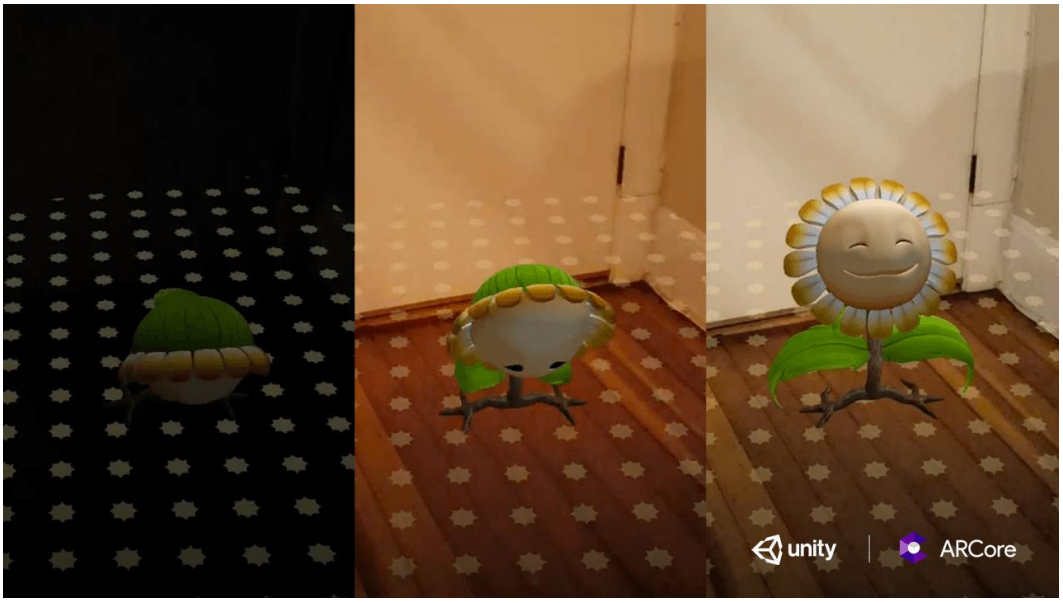


Рис. 3. Light Estimation ARCore

Прив'язки та відстеження

Позиції якорів можуть змінюватися, оскільки ARCore покращує аналіз власної позиції та її оточення. Якщо користувач хоче розмістити віртуальний об'єкт, то йому потрібно визначити якоря, щоб ARCore міг відслідковувати позицію об'єкта з часом. Часто користувач створює якір на підставі позиції дотику, який повертається в ході хіт-тесту.

ARCore може оновлювати положення об'єктів навколишнього середовища, таких як площин та 3D-об’єктів. Площини та точки – це особливий тип об'єкта, який називається відстежуваним. Користувач може прив'язати віртуальні об'єкти до певних трекерів, щоб переконатися, що взаємозв'язок між віртуальним об'єктом і відстежуваним залишається стабільним навіть тоді, коли пристрій рухається навколо. Це означає, що якщо користувач розмістить віртуальний об’єкт на своєму столі, то ARCore налаштовуватиме позицію площини, пов'язаного з робочим столом і таким чином 3D-об’єкт все одно залишиться на вершині стола.

Доповнені зображення

Доповнені зображення дозволяють створювати додатки AR, які можуть відповідати конкретним 2D-зображенням, таким як упаковка продукту або фільму. Користувачі можуть взаємодіяти з доповненою реальністю, коли вони вказують камеру свого телефону на певні зображення, наприклад, вони можуть направити камеру свого телефону на візитку і на ній з’явиться більш детальна інформація.

До зображення можна прив’язати певну інформацію офлайн. Також дозволяється створювати базу даних зображень, з подальшою можливістю додавати об’єкти в реальному часі з пристрою. Після реєстрації ARCore буде виявляти ці зображення, границі зображень та повертати відповідну позицію.

Спільний доступ

ARCore Cloud Anchors API дозволяє створювати спільні або багатокористувацькі додатки для пристроїв Android і iOS.

За допомогою Cloud Anchors один пристрій надсилає прив'язку та розташовує неподалік об'єкт, який вказує на хмару для хостингу. Цими якорями можна поділитися з іншими користувачами на пристроях Android або iOS у тому самому середовищі. Це дозволяє програмам проглядати ті ж самі тривимірні об'єкти, які прикріплені до цих якорів, дозволяючи користувачам одночасно використовувати однакову сцену доповненої реальності.

1.1.2 Apple Arkit

З виходом IOS 11, Apple випустила ARKit API для розробників, які працюють над додатками віртуальної та доповненої реальності. Інструмент ARKit призначений для побудови точної карти оточення за допомогою SLAM. Більш того, для створення доповненої реальності користувачі не потребують зовнішнього обладнання. [2]

З виходом операційної системи IOS 11 усі телефони Apple, починаючи з версії Iphone 6s, отримали підтримку біблотеки ARkit. Слід зазначити, що функціональність ARkit 2.0 підтримують лише телефони Apple на базі операційної системи IOS 12.

ARKit дозволяє максимально використовувати камери та датчики, вже вбудовані в пристрої Apple, для створення програмних застосунків з використанням доповненої та віртуальної реальності.

ARKit надає широкий спектр можливостей у галузі доповненої реальності. Наприклад, програмне забезпечення допомагає ідентифікувати плоску поверхню столу, щоб користувач міг розміщувати на ньому віртуальні об'єкти. Навіть основний досвід роботи з доповненою реальністю, вимагає величезної обчислювальної потужності. Компанія Apple почала стрімко розвивати галузь доповненої реальності і також свою бібліотеку ARKit, щоб створити величезний спектр додатків та ігор, які поєднують віртуальний світ з реальним.

На поточний момент ARkit від Apple є головним конкурентом бібліотеки ARCore від Google.

Розглянемо можливості бібліотеки Arkit:

- Візуальна інерціальна одометрія. ARKit має у своєму складі VIO (Visual Inertial Odometry), щоб правильно аналізувати навколишнє середовище. VIO об'єднує дані, зроблені камерою, з даними CoreMotion. CoreMotion являє собою IMU датчик розроблений компанією Apple. Комбінована інформація дозволяє пристрою точно визначити його рух у приміщенні без будь-якого додаткового калібрування.
- Розуміння сцени та оцінка освітлення. Бібліотека ARKit дозволяє пристрою аналізувати сцену, представлену камерою і визначати горизонтальні поверхні у просторі. Плоскі поверхні підлог і столів, виявлені бібліотекою, можуть бути використані для створення доповненої реальності, коли віртуальні об'єкти можуть бути розміщені на реальній поверхні. Крім того, за допомогою датчиків камери ARKit оцінює освітлення на сцені та використовує цю інформацію для освітлення віртуальних об'єктів.
- Висока продуктивність обладнання та оптимізація рендеринга. Розробники, які використовують ARKit, можуть створювати революційні, інноваційні, привабливі та деталізовані додатки та ігри для додаткової реальності в першу чергу завдяки обчислювальної потужності, наданій процесорами Apple A9, A10 і A11. Функціонування ARKit може бути додатково оптимізоване за допомогою програм SceneKit, Metal або сторонніх програм, таких як Unreal Engine та Unity. На рис. 4 можна побачити приклад роботи ARkit.

Рис. 4. Приклад роботи ARkit

1.1.3 Vuforia

Vuforia – це бібліотека для розроблення програмного забезпечення для мобільних пристроїв, що дозволяє створювати додатки з використанням доповненої реальності. Vuforia використовує технологію комп'ютерного зору для розпізнавання та відстеження планарних зображень (графічних об'єктів) та простих 3D-об'єктів, таких як коробки, в режимі реального часу. Можливість реєстрації зображень дозволяє розробникам позиціонувати та орієнтувати віртуальні об'єкти, такі як 3D-моделі та інші, з зображеннями реального світу, коли вони переглядаються через камеру мобільного пристрою. Віртуальний об'єкт потім відстежує положення та орієнтацію зображення в режимі реального часу так, що перспектива глядача на об'єкті відповідає перспективі на зображення маркеру. Таким чином, видається, що віртуальний об'єкт є частиною реальної сцені. На рис. 5 можна побачити приклад роботи Vuforia. [3]

Рис. 5. Приклад роботи Vuforia

SDK Vuforia підтримує цілий ряд 2-х та 3-мірних цільових типів, включаючи цілі зображення без маркеру, тривимірні 3D-конфігурації та форму адресного маркеру відомих як VuMark. Додаткові можливості SDK включають в себе локалізоване виявлення оклюзії за допомогою віртуальних кнопок, вибір цільових зображень та можливість створення та переналаштування маркерів програмним способом під час виконання.

Vuforia надає інтерфейси програмування додатків (API) на C ++, Java, Objective-C ++ (мова, що використовує комбінацію синтаксису C ++ та Objective-C), і мови .NET за допомогою розширення для рушія Unity. Таким чином, SDK підтримує як нативну розробку для iOS і Android, так і для розроблення AR-додатків в Unity, які легко переноситься на обох платформах. Програми AR, розроблені з використанням Vuforia, є сумісними з широким спектром мобільних пристроїв, включаючи телефони та планшети iPhone, iPad та Android, які працюють на ОС Android версії 2.2 або новішої, а також процесорів ARMv6 або 7 з можливостями обробки FPU (Floating Point Unit).

Функції розпізнавання та відстеження бібліотеки Vuforia можуть використовуватися для різних зображень та об'єктів таких як:

1) Маркерні моделі: дозволяють розпізнавати об'єкти за формою за допомогою вже існуючих 3D-моделей. Дозволяють замістити вміст AR на найрізноманітніших предметах, таких як

промислові пристрої, транспортні засоби, іграшки та побутові прилади.

2) Зображення маркерів: дозволяють додавати вміст на плоскі зображення, наприклад, друковані носії та упаковку продуктів.
3) Об'єкти-маркери: створюються шляхом сканування об'єкта. Вони є гарним вибором для іграшок та інших виробів з детальною деталізацією поверхні та стабільною формою.
4) Багатоцільові маркери: створюються з використанням кількох об'єктів зображення та можуть бути розташовані в звичайних геометричних фігурах (наприклад, коробках) або в будь-якому довільному розташуванні плоских поверхонь.
5) Маркери для циліндрів: зображення на об’єктах, що мають приблизно циліндричну форму (наприклад, пляшки для напоїв, чашки для кави, банки з содою).
6) VuMarks: це налаштовані маркери, які можуть кодувати ряд форматів даних. Вони підтримують унікальну ідентифікацію та відстеження для програм AR.
7) Зовнішня камера: доступ до відеоданих з камери поза межами телефону або планшета під час створення доповненої реальності.
8) Ground Plane: дозволяє розмістити вміст на горизонтальних поверхнях у середовищі, таких як столи та підлоги.
9) Extended tracking – це особлива функціональна особливість, яка надає змогу бачити об’єкт доповненої реальності навіть при загубленні маркеру. Розширене відстеження використовує IMU пристрій для покращення ефективності відстеження та підтримання відстеження, навіть якщо ціль більше не в зоні дії камери.

1.1.4 Microsoft Hololens

Microsoft HoloLens - окуляри змішаної реальності, розроблені компнією Microsoft. Окуляри використовують 64-розрядну операційну систему Windows Holographic (версія Windows 10).

HoloLens представляє собою обруч з розташованими перед очима тонованими лінзами з хвилеподібною призматичною структурою, які заломлюють і відправляють в очі користувача зображення з розташованих з боків мікродісплеев. Для використання HoloLens має бути відкаліброване міжзрачкова відстань. Розмір пристрою може бути пристосований під розмір голови користувача за допомогою спеціального коліщатка. У верхній частині розташовані 2 пари кнопок - для управління яскравістю екрану (над лівим вухом) і гучністю звуку (над правим). Сусідні кнопки мають різну форму (одна опукла, інша увігнута) з тією метою, щоб їх можна було розрізняти на дотик. Динаміки розташовані у нижньому краю пристрою дозволяють чути як звуки віртуальної реальності, так і звуки, які виходять зовні. [4]

1.2 Аналіз існуючих способів генерації освітлення

У цьому підрозділі розглянемо існуючі способи та алгоритми генерації освітлення, які використовуються у технологіях доповненої реальності.

1.2.1 Алгоритм вирізання тіней

Даний алгоритм надає змогу виявляти тіні на зображенні. Алгоритм визначає тіні, освітлені області та однакові поверхні у сцені з використанням інваріантної відстані від вимірювання освітленості. Ці площі використовуються для оцінки параметрів афінної моделі освітлення тіні. Як результат, можна отримати карту тіней у просторі, яку можна використати для якісного освітлення 3D об’єкту в неосвітленних місцях реального простору. Одною з основних вимог алгоритму є якісне

зображення, яке подається на вхід алгоритму. На рис. 6 показано приклад роботи даного алгоритму. [5]

Рис. 6. Приклад роботи алгоритму вирізання тіней. Зліва – оригінальне зображення, а праворуч – з використанням алгоритму вирізання тіней

1.2.2 Алгоритм Light estimation

Даний алгоритм налаштовує освітлення сцени доповненої реальності на основі аналізу відеопотоку з задньої камери смартфону. На поточний момент є декілька модифікацій алгоритму, розроблених компаніями Apple та Google. На рис. 7 наведено діаграму дій роботи стандартного алгоритму light estimation. [6]

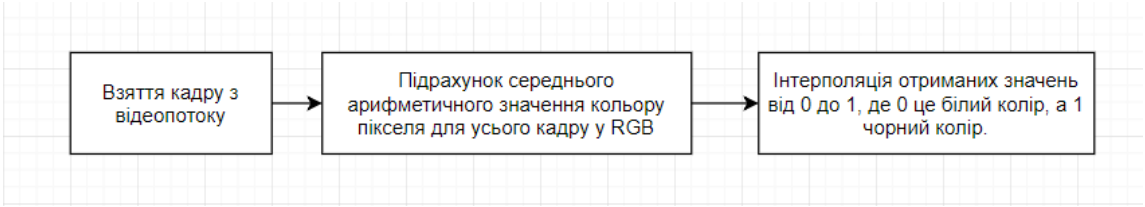


Рис. 7. Діаграма дій роботи алгоритму light estimation.

На основі отриманого значення результату роботи алгоритму, змінюється значення інтенсивності освітлення кожного об’єкту світла у сцені доповненої реальності. Також є модифікація алгоритму, розроблена компанією Google. На рис. 8 наведено діаграму дій роботи модифікованого алгоритму light estimation.

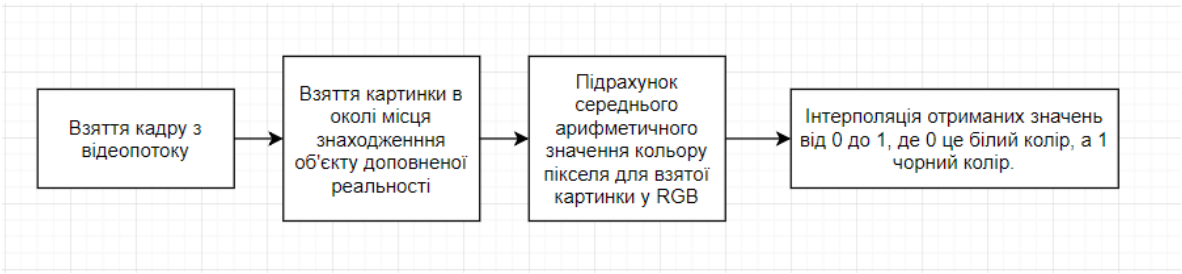
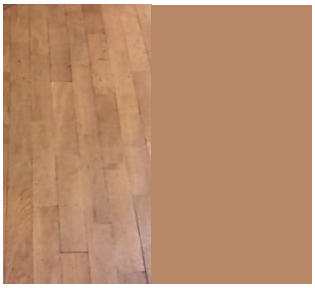


Рис. 8. Діаграма дій модифікованого алгоритму light estimation

Модифікований алгоритм light estimation відрізняється від стандартного тим, що для обробки алгоритму буруть не весь кадр, а частину кадру в околі місця знаходження об’єкту допоненої реальності. На рис. 9 наведено приклад взяття частини кадру в околі місця знаходження 3D-об’єкту. На рис 10 показано обчислене середне значення кольору для результуючої частини зображення.

а) б) в) г)

Рис. 9. Приклад взяття частини кадру для модифікованого алгоритму Light estimation, де: а) – це зображення з 3D-об’єктом доповненої реальності, б) – це зображення з найденим околом 3D-об’єкту, в) – це зображення без 3D-об’єкту, г) – це зображення без 3D-об’єкту з результуючим околом, який буде використаний у алгоритмі.



а) б)

Рис. 10. Результат взяття середнього арифметичного значення кольору для знайденої частини кадру, де а) – це частина кадру зображення, яка була сформована околом 3D-об’єкту, а б) – це середне арифметичне значення кольору а).

Оскільки результатом алгоритму повинно бути число від 0 до 1, то над результатом середнього значення кольору, який у наведеному прикладі дорівнює вектору (182, 136, 104) RGB, потрібно провести інтерполяцію до значення від 0 до 1, де 0 – це вектор (0, 0, 0), а 1 – це вектор (255, 255, 255). З цього маємо наступне (182 + 136 + 105) / (255 * 3) = 0.55. Це означає що освітлення сцени доповненої реальності повинно бути виставлено на 55% від максимального значення. Можна виділити наступні переваги і недоліки алгоритму Light Estimation.

Переваги:

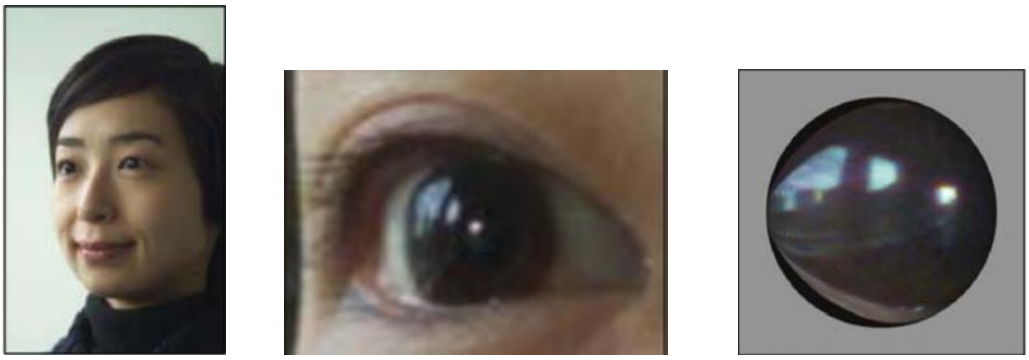
- Проста реалізація алгоритму.
- Висока швидкодія алгоритму.
- Можливість бути сумісним з будь-яким смартфоном.
- Низькі вимоги до апаратної частини девайсу.

Недоліки:

- Неякісні результати.
- Відсутність можливості генерації навколишнього освітлення.
- Відсутність можливості генерації тіней.

1.2.3 Алгоритм Eye light probe

Алгоритм Eye light probe використовує око людини для аналізу освітлення в приміщенні. Даний алгоритм використовується у різних способах захисту інформації або аутентифікації користувача. Основна ідея алгоритму полягає у тому, щоб визначити місцезнаходження ока коритувача на картинці і використати цю частину картинки як карту освітлення. На рис. 11 наведено результат роботи алгоритму. [7]



а) б) в)

Рис. 11. Приклад роботи алгоритму Eye light probe, де: а) вхідне зображення з камери, б) знайдене око, в) результат створення карти відбиття на основі зображення ока.

Можна виділити наступні переваги і недоліки алгоритму Eye light probe.

Переваги:

- Висока якість при ідеальних умовах.
- Можливість створення детальної карти освітлення.

Недоліки:

- Складність у знаходженні очей при недостатньому освітленні.
- Низька якість алгоритму при недостатньому освітленні.
- Можлива низька якість роботи алгоритму при поганій якості камери.

1.3 Висновки до розділу

У даному розділі були розглянуті існуючі рішення для взаємодії з доповненою реальністю з різноманітними функціональними особливостями. Оскільки на момент початку розробки програмного забеспечення, у доступності є лише смартфон від компанії Apple, то було обрано взаємодіяти з доповненою реальністю на основі функціональних особливостей бібліотеки бібліотек ARKit.

Були розглянуті алгоритми генерації освітлення та проаналізовані переваги та недоліки цих алгоритмів.

Аналіз показав, що наявні рішення для взаємодії з доповненою реальністю, можна побачити, що всі вони мають або не мають власні алгоритми обробки освітлення.

Всі розглянуті рішення не мають якісного алгоритму генерації тіней та освітлення, можна зробити висновок, що на поточний момент є необхідність створення вдосконаленого способу генерації освітлення і цей спосіб варто основувати на бібліотеках ARkit та ARCore. Слід зазначити, що спосіб розроблений на основі бібліотеки Arkit, може бути легко портований під пристрої на основі операційної системи Android, завдяки подібності бібліотек ARkit та ARCore.

РОЗДІЛ 2 ВДОСКОНАЛЕНИЙ СПОСІБ ГЕНЕРАЦІЇ ОСВІТЛЕННЯ ТА ТІНЕЙ ДЛЯ ОБ'ЄКТІВ ДОПОВНЕНОЇ РЕАЛЬНОСТІ

2.1 Математичні засади реалізації модифікованого способу генерації освітлення та тіней об'єктів доповненої реальності

У цьому підрозділі розглянемо основні математичні засади, які будуть використовуватись при розробленні способу генерації освітлення та тіней об'єктів доповненої реальності.

2.1.1 Сферичні гармонійні коефіцієнти

Сферичні функції являють собою кутову частину сімейства ортогональних рішень рівняння Лапласа, записану в сферичних координатах. Вони широко використовуються для вивчення фізичних явищ в просторових областях, обмежених сферичними поверхнями, а також при вирішенні задач, пов'язаних із фізичними явищами зі сферичної симетрією. Сферичні функції застосовуються у теорії диференціальних рівнянь та у теоретичній фізиці, зокрема для розв'язку задач обчислення електронних орбіталей в атомі, гравітаційного поля геоїда, магнітного поля планет та інтенсивності реліктового випромінювання. [8]

В математиці сферична гармоніка – це кутова частина ряду розв'язків рівняння Лапласа, представлена у сферичних кординатах. У 3D комп'ютерній графіці сферичну гармоніку використовують для стоворення карти освітлення, карти тіней і для моделювання 3D об'єктів. Сферичні гармоніки позначаються як $Y_{l,m}(\theta,\varphi)$ та обчислюються наступним чином:

$$Y_{l,m}(\theta,\varphi) = (-1)^{\frac{m+|m|}{2}} \sqrt{\frac{2l+1}{4\pi} * \frac{(l-m)!}{(l+m)!}} P_l^{|m|}(cos\theta) e^{im\varphi} \qquad (1)$$

де $P_l^{|m|}(x)$ – це приєднані поліноми Лежандра, l = 0, 1, 2… , а m може приймати значення від -1 до 1.

У рушії Unity сферичні гармонійні коефіцієнти використовуються для створення Light Probe Group та SkyLight, тобто карти освітлення сцени.

Сферичні гармонійні коефіцієнти нерідко застосовуються для вирішення проблем захисту інформації або аутентифікації користувача. На сьогодні існує багато бібліотек комп'ютерного зору, таких як OpenCV або ARkit, які забезпечують цю функціональність. На рис. 12 наведено приклад використання сферичних коефіцієнтів для створення сферичної карти глибини, яка є однією зі складових аутентифікації користувача за обличчям.

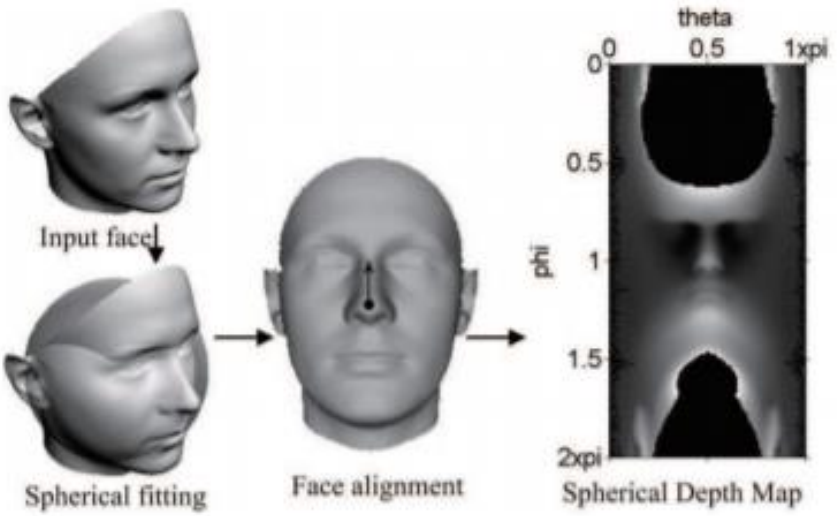


Рис. 12. Розклад обличча у сферичну карту

На рис. 13 представлено сферичні функції $Y_{l,m}$, де l = 0 ... 4 (рахуючи зверху вниз), m = 0 ... 4 (рахуючи зліва направо). Функції негативного порядку Y_{l-m} повернені навколо осі Z на $\frac{90}{m}$ градусів відносно функцій позитивного порядку.

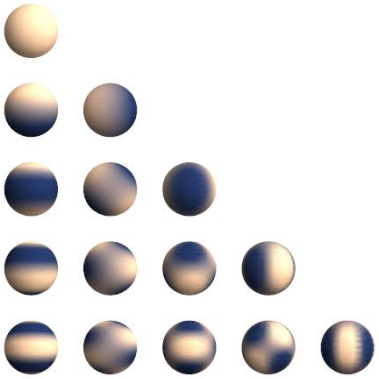


Рис. 13. Репрезентація сферичних гармонік Лапласа

2.1.2 Розмиття Гауса

При застосуванні розмиття Гауса (також відомого як згладження Гауса) до об’єкту, що оброблюється, відбувається його розмиття за Гаусовою функцією. Цей ефект найчастіше використовується при розробленні графічного програмного забезпечення для зменшення шумів зображення та зменшення деталізації. Результатом застосування цієї функції є гладке розмиття, яке нагадує проглядання зображення через прозорий екран, що кардинально відрізняється від ефекту боке, який створюється об'єктивом камери поза фокусом. Гаусове згладжування також використовується як етап попередньої обробки алгоритмів комп'ютерного зору для покращення структури зображення в різних масштабах. Нище наведено формулу згладження Гауса для двовимірного простору.

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \qquad (2)$$

де x та y це координати точок, а σ – це стандартне відхилення розподілу Гауса.

На рис. 14 наведено приклад згладжування зображення із розмиттям за методом Гауса.

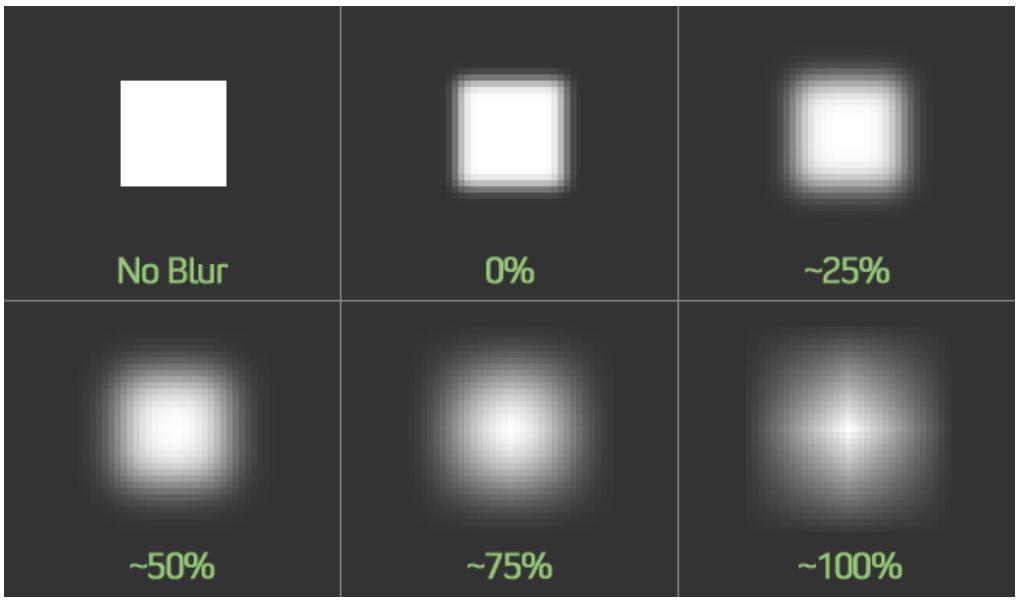


Рис. 14 Розмиття Гауса

2.1.3 Використання датчику освітлення

Датчик зовнішнього освітлення є складовою частиною смартфонів, ноутбуків, інших мобільних пристроїв, автомобільних дисплеїв та ЖК-телевізорів. Він являє собою фотоприймач, який використовується для заміру кількості навколишнього світла та відповідного зменшення яскравості екрана пристрою. Це надає змогу динамічно змінювати інтенсивність яскравості екрану електронного пристрою залежно від ступіня освітлення навколишнього середовища. З практичної точки зору затемнення екрана значно продовжує термін служби акумулятора.

Стандартна міжнародна одиниця виміру ступіня освітлення навколишнього світла - люкс. Типова характеристика датчика зовнішнього освітлення становить менше ніж 50 люкс у сутінках та до 10 000 люксів при денному освітлені.

Існує три типи датчиків зовнішнього освітлення: фототранзистори, фотодіоди та фотонні інтегральні схеми, які об'єднують фотоприймач та підсилювач в одному пристрої.

У 2004 році 30% смартфонів мали вбудований сенсор зовнішнього освітлення, в 2016 році ця цифра зросла до 85%.

Датчик освітлення дозволяє додатково корегувати загальне освітлення сцени у доповненій реальності. На рис. 15 представлене найчастіше місце знаходження датчику освітлення.

Рис. 15. Датчик освітлення телефона

2.2 Обґрунтування способу генерації освітлення і тіней об’єктів доповненої реальності

Пропонований спосіб генерації освітлення ґрунтується на алгоритмі Light Estimation. Цей алгоритм забезпечує зміну освітлення всієї сцени, однак в певних ситуаціях це може давати неякісний результат, який не буде прийнятним для застосування в деяких умовах навколішнього середовища. Для подолання цієї проблеми був запропонований альтернативний спосіб генерації освітлення.

Для роботи пропонованого алгоритму було використано стандартний датчик освітлення.

Пропонований спосіб складається з двох частин: генерації карти освітлення та додаткове затемнення об’єкту, якщо він знаходиться у тіні.

Розглянемо першу частину алгоритму – генерацію карти освітлення.

На першому кроці на вхід подається зображення обличчя користувача, отримане із використанням фронтальної камери смартфону, після чого це зображення перетворюється у зображення у градації сірого. У лістингу 1 наведено програмний код для перетворення зображення у градації сірого.

Лістинг 1

```
public static Texture2D Grayscale(Texture2D inTex)
    {
        return Filter(inTex, GrayscaleFunc);
    }

private static void GrayscaleFunc(int width, int height, Color[] inPixels,
ref Color[] outPixels, params object[] parameters)
    {
        for (int i = 0; i < inPixels.Length; i++)
        {
            float gray = inPixels[i].grayscale;
            outPixels[i] = new Color(gray, gray, gray, inPixels[i].a);
        }
    }
```

Результатом цього етапу є зображення обличчя користувача у градації сірого, яке було збережене у об’єкт типу Texture2D.

На наступному етапі на отриманому зображенні визначаються межі обличчя користувача шляхом застосування інструменту Arkit, який, як правило, використовується у методах захисту інформації та аутентифікації користувача. Зокрема, на цьому етапі застосовується функція ARkitFaceFind(). Для реалізації кроку на Android та ARCore можуть бути використані можливості бібліотеки OpenCV, де також реалізована ця функціональна особливість.

В результаті отримуємо окіл обличчя, як це показано на рис. 16.

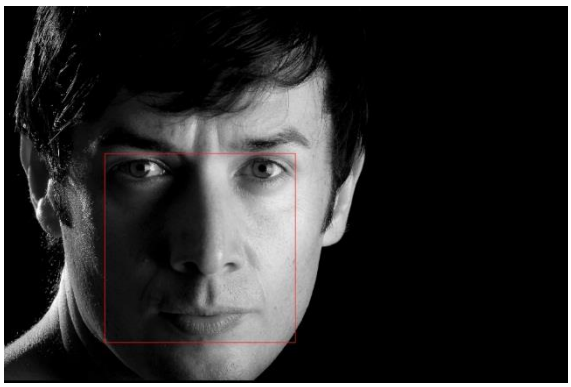


Рис. 16. Зображення обличчя у градації сірого зі знайденим околом обличчя

Arkit надає наступну інформацію про місцезнаходження знайденої границі обличчя:

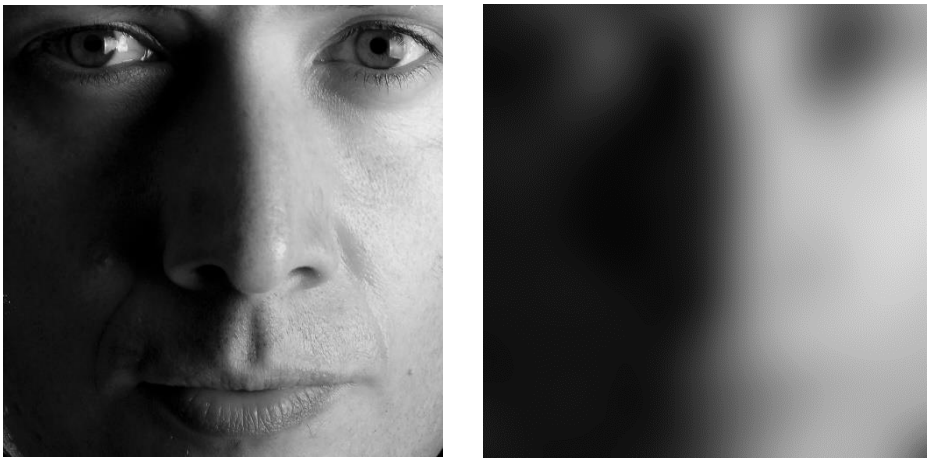
- координати лівого верхнього краю границі представлені як Vector2;
- цілочисельне значення ширини границі у пікселях;
- цілочисельне значення висоти границі у пікселях.

На основі отриманих даних створюється нове зображення типу Texture2D. За допомогою функції Color[] Texture2D.GetPixels(int x, int y, int blockWidth, int blockHeight, int miplevel) виокремлюється отриманий фрагмент зображення та записується у нову текстуру заданої ширини та висоти за допомогою функції SetPixels(Color[] colors, int miplevel = 0). В результаті на виході створюється нова текстура заданої ширини та висоти, приклад якої наведений на рис. 17.

На наступному кроці над текстурою виконується операція розмиття, або blur. Операція розмиття виконується за допомогою функції Гауса, на основі якої був створений шейдер, що здійснює операцію розмиття. Використанн саме цього шейдеру є принциповим, оскільки процес його виконання відбувається на графічному процесорі пристрою, що забезпечує швидкість обчислення поставленої задачі. Після операції розмиття текстури шейдером, текстура записується у нову текстуру, результат якої наведено на рис. 17.

На четвертому етапі відбувається налаштування матеріалу для карти освітлення. З цією метою був створений матеріал Unity з шейдером Skybox/Panoramic, який на вхід приймає текстуру, а на виході генерує сферичну карту.

Експериментально на різних наборах вхідних даних було визначено параметри матеріалу для карти освітлення. У таблиці 1 наведені результати проведених експериментів – визначені параметри налаштування для створеного матеріалу MyNewLightMap.mat для карти освітлення.



а) б)

Рис. 17. Результат розмиття текстури шейдером, де а) вхідна текстура, а б) результат обробки текстури шейдером

Таблиця 1 – Значення параметрів матеріалу карти освітлення

Параметр	Значення
Tint Color	RGB (75,75,75)
Exposure	0.82
Rotation	CameraRotation
Mapping	Latitude Longitude Layout
Image Type	360 Degrees
Spherical (HDR)	Input Texture

Створений матеріал має вхідний параметр посилання на текстуру Spherical (HDR), яку потрібно оновлювати при обробленні нового зображення обличчя.

Параметр Exposure відповідає за силу освітлення білої частини згенерованої карти освітлення, а параметр Rotation за поворот карти у просторі. Параметр Exposure був штучно підібраний, щоб отримати найкращу якість, в той час як параметр Rotation обчислюється відповідно до повороту пристрою у навколишньому середовищі. Оскільки ARkit

оброблює позицію та поворот головної камери сцени, то значення параметру Rotation береться від поточного значення повороту камери. На рис. 18 наведено вигляд згенерованої карти у розгорнутому вигляді.

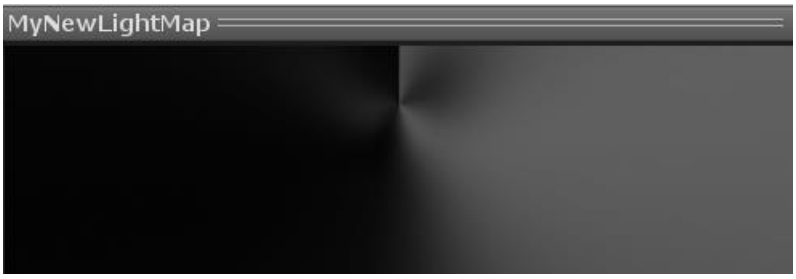


Рис. 18. Згенерована карта освітлення у розгорнутому вигляді

Наступним кроком генерується карта освітлення (матеріал сферичної карти), яка повинна бути присвоєна до загальної карти освітлення SkyboxMaterial об’єкту загального освітлення Environment Light.

На рис. 19 наведено параметри налаштування світла сцени у рушії Unity.

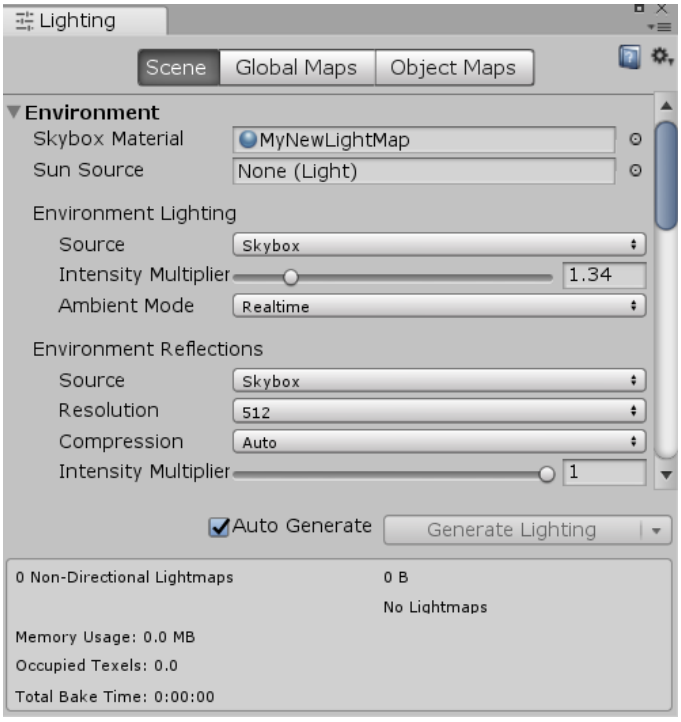
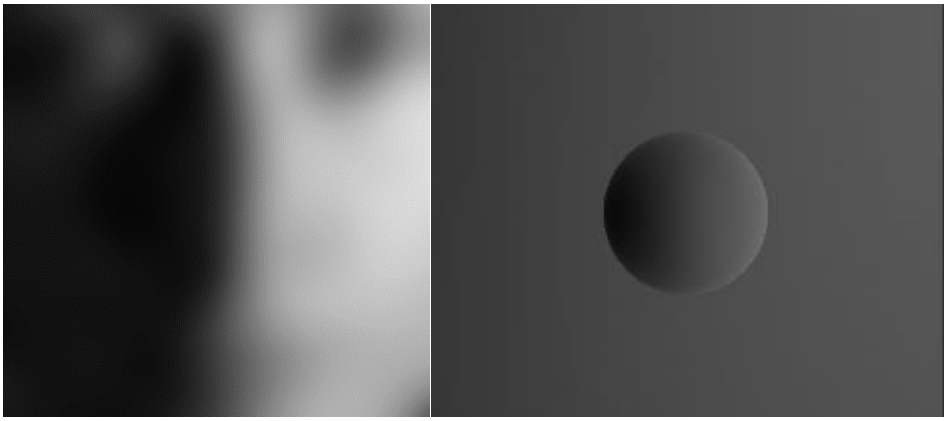


Рис. 19. Налаштування загального освітлення сцени.

Одним з найважливіших параметрів налаштувань загального освітлення є параметр Intensity Multiplier, який відповідає за інтенсивність освітлення створеної карти. Оскільки налаштування сцени вже мають посилання на створений матеріал, то будь-які зміни матеріалу приведуть до зміни Skybox. Відповідно, якщо на вхід прийде нове зображення, зміна параметру текстури матеріалу MyNewLightMap.mat призведе до зміни освітлення всієї сцени. На рис. 20 наведено вид сцени з налаштованою картою освітлення, яку було взято з обличчя, зображеного на рис. 16.



а) б)

Рис. 20. Освітлення сфери згенерованою картою освітлення, де а) згенерована карта освітлення, а б) вид сцени Unity з використанням згенерованої мапи освітлення та 3D-об’єктом сфери

На наступному етапі визначаємо напрямок головного світла створеної сферичної карти освітлення, використовуючи змінну LightmapData.lightmapDir. У цій змінній зберігається Vector3 повороту домінуючого світла у створеній карті освітлення. На рис. 21 наведено зображення сфери, яка була освітлена за допомогою створеної карти освітлення з різних ракурсів. Як видно з ілюстрації, домінуючий напрямок буде протилежним осі x, що дорівнюватиме Vector3(0,-90,0) повороту освітлення спрямованого світла.

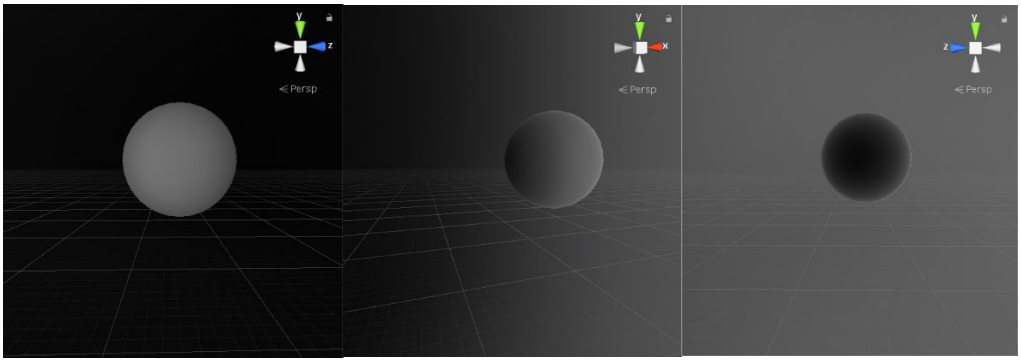


Рис. 21. Освітлення сфери з різних ракурсів

На наступному кроці відбувається перехід до сцени виставлення об’єктів доповненої реальності. Сама сцена базується на тестовій сцені, яка йде в навчальних матеріалах з плагіном ARkit.

Розглянемо тепер другу частину алгоритму – додаткове затемнення об’єкту, якщо він знаходиться у тіні, що є модифікацією алгоритму Light Estimation.

У розділі 1 пункту 1.2.2 було описано схему роботи алгоритму Light estimation. Алгоритм Light estimation видає значення від 0 до 1, на значення якого змінюються всі інтенсивності об’єктів типу Light у сцені. Оскільки розроблений спосіб використовує карту освітлення, то було прийнято рішення модифікувати алгоритм Light estimation. Також слід пам’ятати про те, що згенерована карта освітлення вже освітлює 3D-об’єкти з якісним освітленням, тож з алгоритму Light estimation потрібно вилучити лише функціональну особливість затемнення сцени при знаходженні 3D-об’єкту на місці тіні. Для того, щоб розділити дану функціональність був використаний датчик освітлення телефону. Спочатку потрібно провести інтерполяцію значення отриманого з датчику освітлення. У таблиці 2 наведені значення lux до рівня освітленості.

Таблиця 2 – Відповідність значення lux до різноманітних умов

0.0001	Світло Сиріуса, найяскравішої зірки нічного неба
0.002	Безмісячне зоряне небо зі світінням
0.01	Чверть Місяця
0.27–1.0	Повний місяць у ясну ніч
1	Повний місяць в тропіках
до 20	У морі на глибині ~ 50 м
80	Житлова кімната
100	Дуже темний похмурий день
150	Офісне освітлення
320–500	Схід та захід сонця в ясний день
400	Похмурий день, типове освітлення у кіностудії
1000	Ясний сонячний день (в тіні)
10,000–25,000	Пряме сонячне світло

Значення з пристрою інерполюється, де 0 – це 100 lux, а 1 – це 20000 lux. Коли відбувається рух об’єкта доповненої реальності, то виконується наступна перевірка. Якщо різниця значення отриманого з датчику освітлення та результату роботи алгоритму Light estimation більше або дорівнює 0.25 (значення отримано експериментальним шляхом), то відбудеться зменшення значення Intensity Multiplier у карті освітлення Skybox на значення результату роботи алгоритму Light Estimation. Таким чином алгоритм Light Estimation буде виконуватись тільки за умов, коли 3D-об’єкт доповненої реальності попадає у межі тіні.

Останнім кроком способу є створення тіні на основі згенерованої карти освітлення. Оскільки можна отримати вектор напрямку домінуючого світла з карти освітлення, то можна спів ставити цей напрямок з напрямком об’єкту Directional Light. Directional Light – це компонент напрямленого світла, який може кинути тінь від об’єкта, який він освічує на інший 3D-об’єкт. Для реалізації цього кроку інтенсивність освітлення Directional Light повинна бути виставлена на 0.0001. Для того, щоб світло не освічувало 3D-об’єкт але все-одно мало змогу кидати тінь від 3D-об’єкту на інші 3D-об’єкти. Також був створений шейдер, який надає змогу 3D-об’єктам бути прозорими але все-одно отримувати тінь. Створений шейдер був застосований до матеріалу об’єкту plane (2D-площина), який прив’язаний до точки місцезнаходження об’єкту, який буде демонструватися у доповненій реальності. На рис. 22 продемонстровано принцип роботи шейдеру.

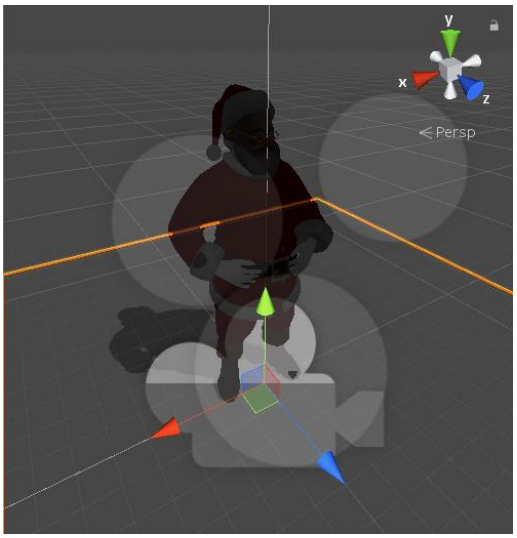


Рис. 22. Приклад роботи шейдеру ShadowOnly.shader

Після виконання останнього кроку, роботу способу можна вважати завершеною. На рис. 23 наведено схему роботи способу генерації освітлення та тіней об’єктів доповненої реальності.

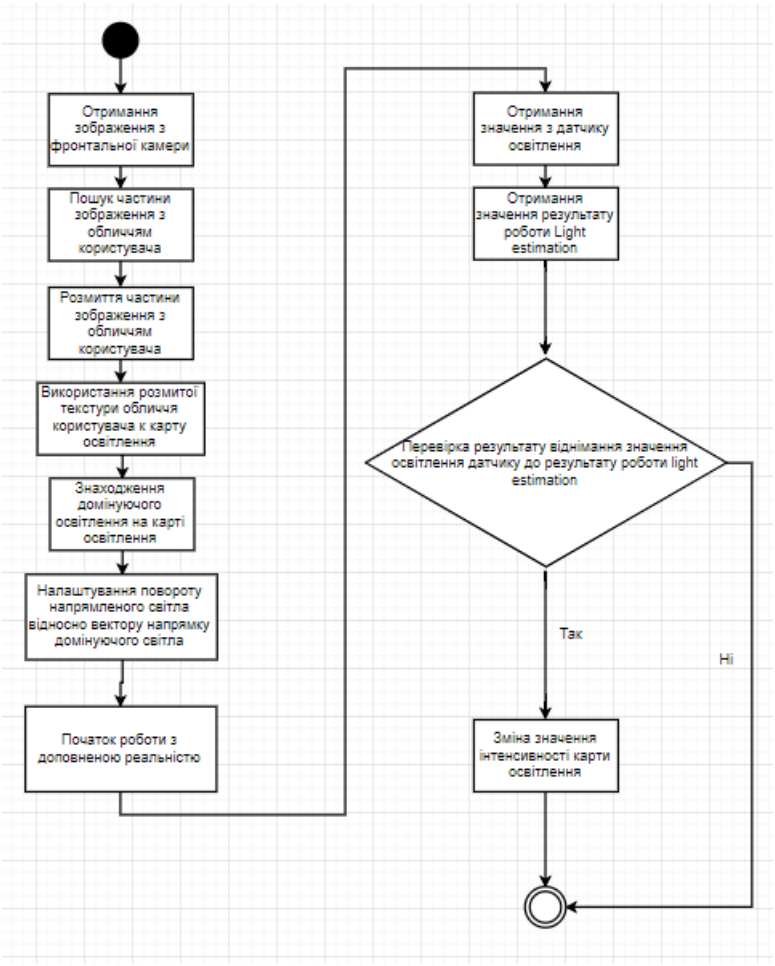


Рис. 23. Схема роботи способу генерації освітлення та тіней об’єктів доповненої реальності

2.3 Висновки до розділу

У цьому розділі наведено модифікований спосіб генерації освітлення та тіней для об’єктів доповненої реальності. Розглянуті математичні засади та особливості алгоритму реалізації способу генерації освітлення та тіней об’єктів доповненої реальності. Додаткове затінення об’єктів пропонується реалізувати на основі модифікованого алгоритму Light Estimation з додатковим використаннм датчику освітлення.

Запропонований спосіб складається з двох частин – генерація карти освітлення та додаткове затінення обробленої сцени доповненої реальності. Наведено апаратні та програмні складові частини пропонованого способу генерації освітлення та тіней об’єктів доповненої реальності.

РОЗДІЛ 3 ОСОБЛИВОСТІ ПРОГРАМНОЇ РЕАЛІЗАЦІЇ СПОСОБУ

3.1 Вибір засобів розроблення програмного забезпечення

Для розроблення програмного забезпезпечення був використаний наступний набір інструментарію:

1. Рушій Unity 2018.2.9f1 – багатоплатформовий інструмент для розробки дво- та тривимірних додатків та ігор.[9]
2. C# – об'єктно-орієнтована мова програмування з безпечною системою типізації для платформи .NET. На поточний час єдина мова, яка підтримується рушієм Unity.[10]
3. Visual Studio 2017 – інтегроване середовище розробки.[12]
4. VirtualBox з встановленим macOS Mojave 10.14.
5. Xcode – інтегроване середовище розробки від Apple. Є важливою складовою у побудові програм під платформу IOS.[11]
6. Windows 10
7. ARKit 2.0 plugin for Unity – бібліотека для взаємодії з доповненою реальністю для платформи IOS. На момент розроблення програмного забезпечення Arkit 2.0 є останню версією бібліотеки.
8. LeanTouch plugin for Unity – найбільш популярний плагін Unity для використання дотикового методу вводу для смартфонів. LeanTouch plugin є обгорткою до стандартних функцій Unity з обробки дотикового методу вводу.
9. iOS Unity plugin– це бібліотека для використання різноманітних функцій пристрою на платформі IOS. Ця бібліотека потрібна для взаємодії з датчиком освітлення телефону.

VirtualBox – це програма віртуалізації для операційних систем, на якій у рамках розроблення програмного забезпечення віртуалізовано операційну систему macOS Mojave.[13]

macOS Mojave 10.14.1 – операційна система macOS від Apple, яка на момент розробки програмного забезпечення є останньою версією операційної системи macOS.

3.2 Архітектура розробленого програмного забезпечення

Оскільки рушій Unity підтримує взаємодію тільки з однією камерою одночасно, то було прийнято рішення розділити програмне забезпечення на дві Unity сцени. У першій сцені відбувається аналіз освітлення з фронтальної камери та предпоказ цього освітлення на об’єкт сфери який прикріплюється до обличчя користувача, а друга сцена надає можливість виставити тривимірний об’єкт у доповненій реальності з освітленням, яке було згенеровано у першій сцені. На рис. 24 наведено ієрархію об’єктів у першій сцені.

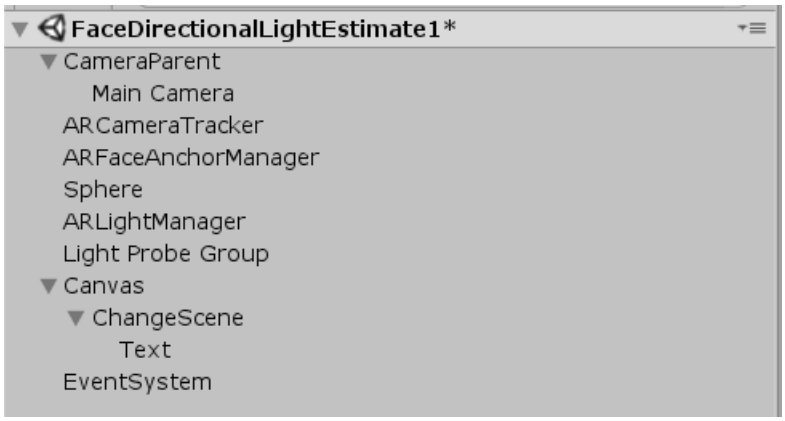


Рис. 24. Ієрархія об’єктів у першій сцені

У цій ієрархії можна побачити об’єкт CameraParent з дочірнім об’єктом MainCamera у якому знаходиться компонент камери, який відповідає фронтальній камері смартфону.

До об’єктів ARCameraTracker та ARFaceAnchorManager прикріплені скрипти ARCameraTracker.cs та UnityARFaceAnchorManager.cs відповідно. Ці скрипти відповідають за знаходження обличчя користувача та зміну положення об’єкту Sphere відносно обличчя користувача.

ARLightManager – це об’єкт на якому знаходиться скрипт ARLightManager.cs. У цьому скрипті виконується основні функції та проходить генерація освітлення, яке потім зберігається в об’єкт LightProbeGroup.

Canvas – це об’єкт в якому знаходиться UI кнопка для переходу на другу сцену.

EventSystem – це об’єкт, який містить у собі компонент EventSystem, який відповідає за обробку подій у сцені. Це обов’язковий компонент для взаємодії з UI елементами сцени.

LightProbeGroup – об’єкт, який містить у собі компонент LightProbeGroup, який відповідає за освітлення сцени. Цей об’єкт зберігає у собі дані освітлення отримані за допомогою створеного способу і має при собі обов’язкову функцію DontDestroyOnLoad() для того, щоб об’єкт не був видалений при переході на другу сцену.

На рис. 25 можна побачити ієрархію та вид другої сцени програмного забезпечення.

Друга сцена також у собі має об’єкт Canvas, який містить у собі кнопку переходу на першу сцену.

CameraParent – це об’єкт, який містить у собі камеру, яка відповідає за основну камеру смартфону.

EventSystem – це такий самий об’єкт як і у першій сцені, що містить у собі компонент EventSystem, який відповідає за обробку подій у сцені. Це обов’язковий компонент для взаємодії з UI елементами сцени.

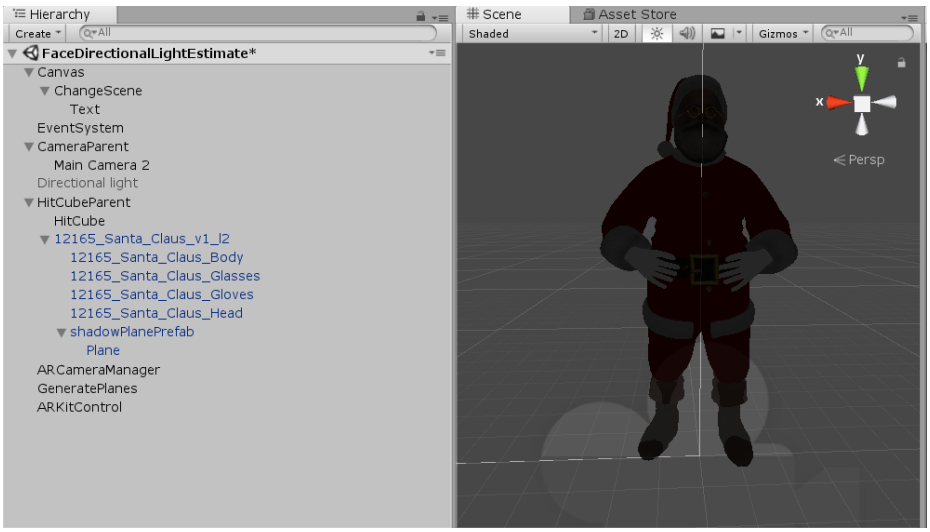


Рис. 25. Ієрархія об’єктів та вид другої сцени програмного забезпечення

DirectionalLight – це об’єкт, який містить у собі компонент спрямованого світла. На рис. 26 можна побачити налаштування компоненту Light. Як можна побачити, інтенсивність світла виставлена на 0.01. Це зроблено для того, щоб світло не освітлювало об’єкт але все одно генерувало тінь.

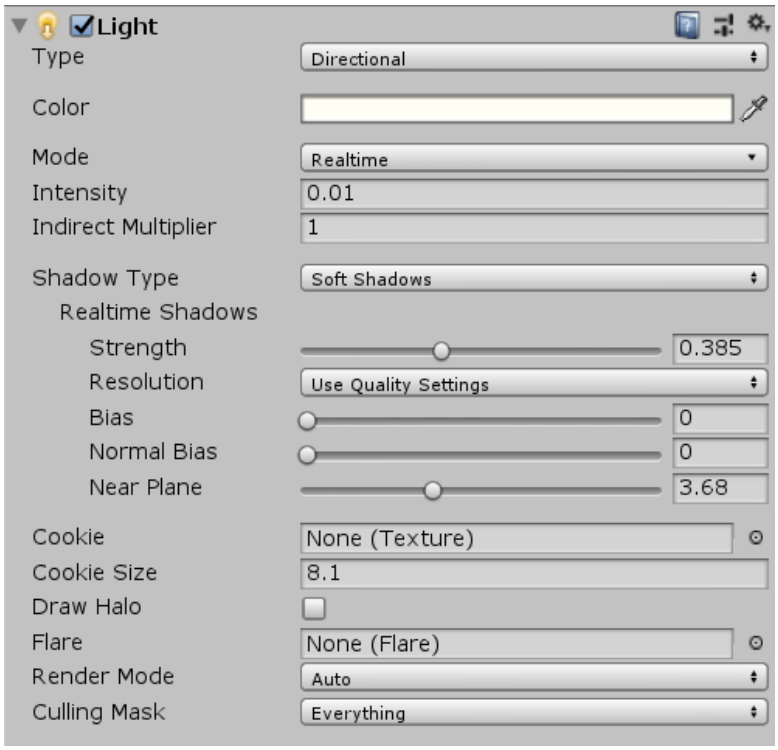


Рис. 26. Налаштування світла

HitCubeParent – це об’єкт, який містить у собі основний меш-об’єкт та об’єкт площини shadowplaneprefab. Об’єкт HitCubeParent це той об’єкт, який буде виставлений у сцену, він має у собі скрипт LeanTranslate з бібліотеки LeanTouch. LeanTouch – це найбільш популярна бібліотека з обробки сенсорного вводу на рушії Unity. ShadowPlanePrefab – це об’єкт, на якому знаходиться власноруч написаний шейдер MobileARShader.shader. Цей шейдер робить площину shadowplaneprefab прозорою з можливістю відображати тільки тіні, які падають на дану площину.

ARCameraManager – це об’єкт, який містить у собі UnityARCameraManager.cs та UnityGenerateLightSecond.cs. Перший скрипт відповідає за налаштування доповненої реальності, а другий відповідає за поворот спрямованого світла відносно світла згенерованого в LightProbeGroup першої сцени, а також обробку даних з датчику освітлення.

GeneratePlanes – це об’єкт, який містить у собі скрипт GeneratePlanes.cs, який відповідає за генерацію площин, на який в майбутньому можна поставити об’єкт HitCubeParent.

Оскільки після побудови Unity проекту під платформу IOS, генерується XCode проект, то потрібно зробити ще пару кроків. Для остаточної побудови Unity проекту під платформу IOS, треба мати доступ до операційної системи macOS Mojave. У моєму випадку операційна система macOS Mojave встановлена на віртуальну машину VirtualBox, також потрібно мати найновішу версію XCode і збудувати згенерований Unity XCode-проект у середовищі XCode на macOS Mojave.

Системні вимоги до програмного забезпечення:

- Iphone 6s та новіше;
- Операційна система IOS 11 та новіше;
- Працездатна фронтальна камера;
- Працездатна основна камера;
- 60 мегабайт вільного місця;

3.3 Висновки до розділу

У цьому розділі обґрунтовано вибір програмних засобів розроблення пропонованого програмного забезпечення. Наведено засоби розроблення програмного забезпечення для генерації освітлення та тіней об’єктів доповненої реальності, а також допоміжні бібліотеки для спрощення поставленої задачі. Описано архітектуру програмного забезпечення, що розроблюється, у якій охарактеризовано загальну структуру програмного засобу та його властивості. Представлено усі бібліотеки та допоміжні програмні застосунки, які були використані при розробленні цього програмного забезпечення. Описано основні частини та функціональність розробленого програмного забезпечення. Наведені обов’язкові програмні застосунки для побудови програмного забеспечення та сформульовані системні вимоги для установки програмного забезпечення на пристрій.

РОЗДІЛ 4 АНАЛІЗ ОТРИМАНИХ РЕЗУЛЬТАТІВ

4.1 Методика оцінювання ефективності способу генерації освітлення та тіней об’єктів доповненої реальності

Оцінка роботи способу та програмного забеспечення буде проводитись на основі наступних критеріїв:

1. Суб’єктивна оцінка:
   - Якість освітлення.
   - Якість тіней.
   - Зручність використання програми.
2. Об’єктивна оцінка тестування:
   - Швидкодія програмного забеспечення.
   - Оцінка показника температури пристрою.

Суперником написаного програмного засобу було обрано влаштований у IOS 12 засіб показу об’єктів доповненої реальності у usdz-форматі.

Файл формату usdz – це новий формат файлу тривимірної сцени, розроблений спеціалістами Apple у співпраці с компанією Pixar. Цей формат файлу надає змогу відображати доповнену реальність через вбудовану програму відображення доповненої реальності на телефонах Apple на базі операційної системи IOS 11 та 12. Цей формат підтримується моделями IPhone 6s та новішими.

На поточний момент usdz файл є еталоном якості у сфері мобільної доповненої реальності, тож було прийнято рішення порівняти створений алгоритм саме з програмним засобом для відображення usdz файлу.

4.2 Тестування розробленого способу генерації освітлення та тіней об’єктів доповненої реальності

Тестування першої сцени розробленого способу відбувалося за умов одного джерела спрямованого світла – сонця. Результати проведеного тесту наведено на рис. 27 – 28.

Тестування другої сцени розробленого способу відбувалося при кімнатній умові. Було протестовано власну програмну розробку та usdz файл. На рис. 29 наведено результати тесту usdz-файлу при кімнатному освітленні. За результатом тесту usdz-файлу, можна побачити, що освітлення моделі відбувається на основі загального освітлення. Такий спосіб освітлення не є якісним, оскільки не може передати освітлення навколишнього середовища.

Рис. 27. Результати освітлення при напрямку світла на обличчя(ліва частина рисунку) та від обличчя(права частина рисунку)

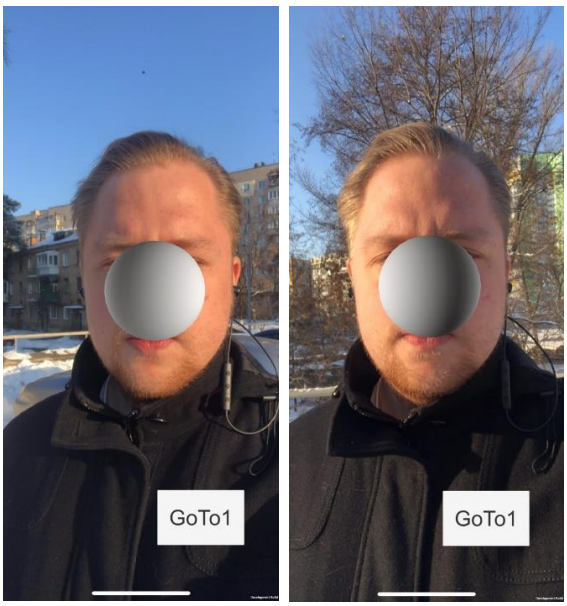


Рис. 28. Результати освітлення при напрямку світла ліворуч від обличчя користувача (ліва частина рисунку) та праворуч від обличчя користувача(права частина рисунку)

Рис. 29. Результати тестування usdz файлу.

На рис. 30 наведені результати розробленої програми, на яких можна побачити збільшення якості при використанні запропонованого способу генерації карти освітлення та тіней для об’єктів доповненої реальності. Загалом об’єкт відрізняється правильним затіненням задньої частини та освітленням передньої частини 3D-об’єкту, а також правильним напрямком

відбивання тіні. На рис. 31 наведено приклад тестового середовища при малому освітленні. В останньому тесті при малому освітленні можна побачити значні переваги розробленого алгоритму на відміну від usdz файлу.

Рис. 30. Результати роботи розробленого алгоритму та програми.

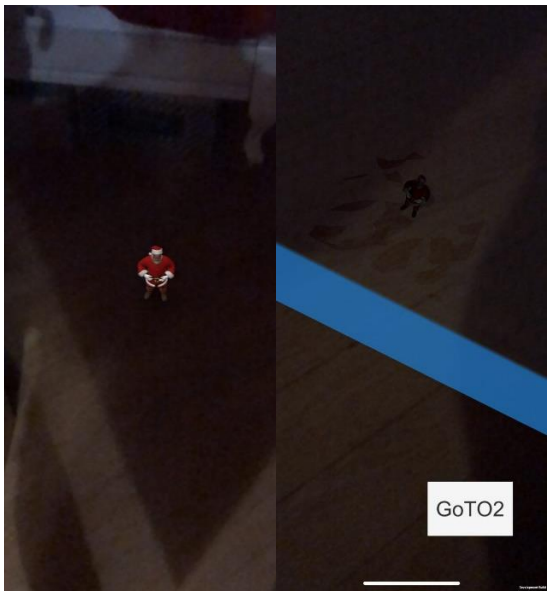


Рис. 31. Результати роботи usdz файлу (ліворуч) та розробленого алгоритму (праворуч) при низькому освітленні.

4.3 Сбір експертної оцінки розробленого способу генерації освітлення та тіней об’єктів доповненої реальності

Розроблений спосіб генерації освітлення та тіней був продемонстрований десятьом експертам, які проаналізували та протестували створений програмний засіб та програмний засіб відображення usdz файлу. У таблиці 3 наведено характеристику обраних експертів.

Таблиця 3 – Характеристики обраних експертів

Порядковий номер експерта	Приблизний вік експерта	Вага експерта (обізнаність у сфері доповненої реальності) від 1 до 10	Зацікавленість експерта у сфері доповненої реальності від 1 до 10
1	20-25	4	5
2	20-25	5	5
3	20-25	3	6
4	20-25	5	3
5	40-50	1	6
6	30-40	9	8
7	15-20	2	10
8	40-50	1	10
9	20-25	10	10
10	20-25	7	3

Експертам було надано доступ до створеного програмного засобу та засобу для перегляду usdz файлу. Тестування відбувалося за різних умов навколишнього середовища. Зібрані результати тестування програмного засобу експертами наведено у таблиці 4. Оцінки створеного програмного

забеспечення наведені у стовпчику ПЗ. Також були надані коментарі від використання обох програмних засобів. На момент тестування програмний засіб для показу usdz файлу працював некоректно, оскільки у IOS 12.1 з’явилися помилки, котрі не давали можливості відображати blob-тінь, то можна побачити, що більшість експертів поставили погану оцінку за якість тіней usdz-файлу але також є декілька експертів у сфері доповненої реальності, які мають знання принципу роботи usdz-файлу у системах IOS 11 та 12.

Таблиця 4 – Оцінки експертів

№ експерта	Якість освітлення Від 1 до 10		Якість тіней Від 1 до 10		Зручність використання програми Від 1 до 10		Коментарі
	ПЗ	usdz	ПЗ	usdz	ПЗ	usdz	
1	6	3	7	4	7	9	—
2	9	5	6	3	9	8	—
3	8	7	8	1	7	9	—
4	7	7	5	1	5	7	—
5	9	1	6	1	6	9	—
6	8	5	10	4	5	7	У розробленій програмі набагато більше кроків до показу моделі у доповненій реальності ніж у usdz файлу
7	6	4	10	1	4	8	—
8	10	3	10	1	6	9	—

Продовження таблиці 4

9	9	7		9	6	7	10	Освітлення генерується коректо але usdz набагато зручніший у використанні
10	6	4		6	8	5	10	Неправильний напрямок тіні

Згідно з отриманими результатами експертизи, було підраховано відсоток удосконаленості розробленої програми відносно формату usdz. Результати аналізу наведено у таблиці 5.

Таблиця 5 – Загальна оцінка способів

	Розроблене програмне забезпечення	Програмний засіб показу usdz-файлів	Відсоток удосконалення
Якість освітлення	78	46	69%
Якість тіней	77	30	156%
Зручність використання	61	86	-29%
Загалом	216	162	33%

На основі отриманих результатів можна зазначити, що розроблене програмне забезпечення суб’єктивно працює на 33% краще за програмний засіб для показу usdz-файлів. Слід зазначити, що розроблений програмний засіб програє програмі відтворення usdz-файлів у зручності використання але суттєво виграє у якості освітлення та тіней. Оскільки, на час розробки програмного забезпечення увагу було акцентовано саме на збільшенні

якості генерації тіней та освітлення, то недоліки у критерії зручності використання були очікувані.

4.4 Об’єктивна оцінка тестування розробленого способу

Важливою складовою тестування програмного забезпечення є оцінка його швидкодії. Для програмного забезпечення, яке працює з 3D-об’єктами, оцінка швидкодії відбувається після аналізу FPS.

Для розробленого програмного забезпечення генерації освітлення та тіней об’єктів доповненої реальності та програмного засобу для показу usdz-файлу був проведений тест FPS на малому інтервалі часу. Результати проведеного тесту можна побачити на діаграмі 1.

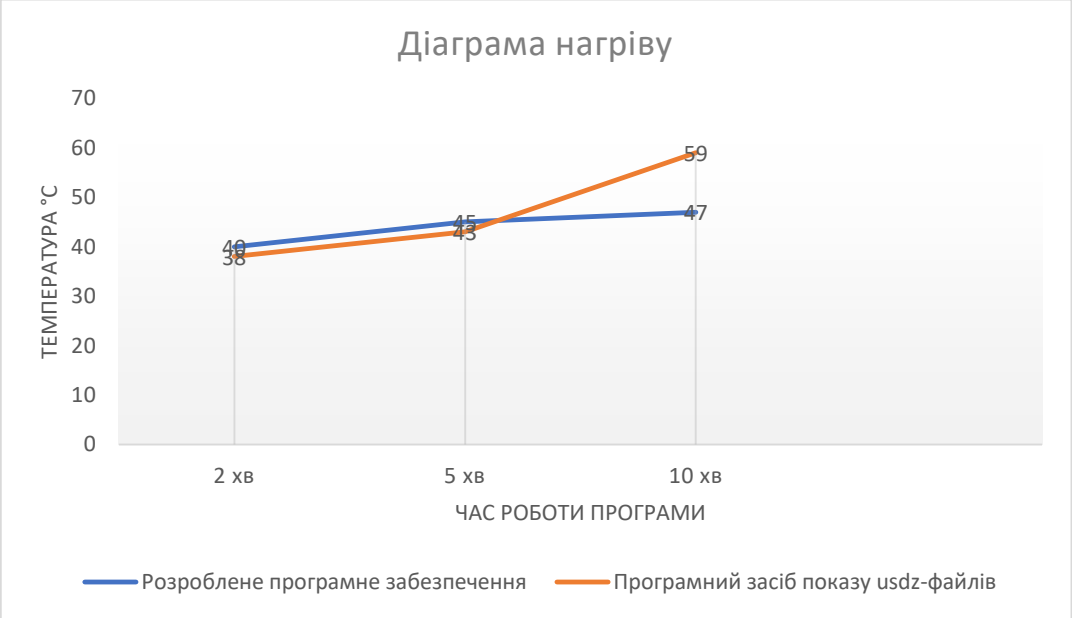
Діаграма 1

Для аналізу швидкодії був взятий інтервал у 10 секунд. На цьому інтервалі було отримано кількість кадрів за секунду для кожної програми. Як можна побачити на діаграмі, значення кадрів за секунду розробленого програмного забезпечення не опускалось нище відмітки у 60 кадрів/сек. Це може бути пов’язано з якісною оптимізацією Unity при обробці 3D-об’єктів, а також оптимізованим модифікованим способом генерації освітлення та

тіней. Оскільки при побудові програми під платформу IOS автоматично включається вертикальний синхроімпульс, то значення FPS не може перевищувати 60.

Оцінка показника температури пристрою проводилася, згідно результатів отриманих з безконтактного термометру. Цей показник є дуже важливим для кінцевого споживача, оскільки великий нагрів пристрою, говорить про те, що програма є неоптимізована і таким чином батарея пристрою може швидко дуже швидко розряджатися. Слід зазначити, що нагрів пристрою при використанні доповненої реальності є природною поведінкою пристрою. Нагрів пристрою при використанні програм з використанням доповненої реальності, був зазначений у специфікації апарату. На діаграмі 2 можна побачити замір нагріву смартфону відповідно до часу використання програм.

Діаграма 2



Для аналізу нагріву пристрою було взято три проміжки у 2, 5 та 10 хвилин. Проаналізувавши отримані дані, можна побачити, що на проміжку до 5 хв температура пристрою менша ніж у програмному засобі показу usdz-файлів в середньому на 2 °С. Слід зазначити, що при довгій роботі програми

(до 10 хв) температура пристрою при роботі розробленого програмного забезпечення набагато менша ніж у програмного засобу показу usdz-файлів. Аналіз показав, що на проміжку до 5 хв розроблена програма нагріває апарат на 5% сильніше за програмний засіб показу usdz-файлів але на 20% менше на проміжку до 10 хв. Середнє значення удосконалення оцінки показника температури пристрою, створеного програмного забезпечення становить 15%.

4.5 Висновки до розділу

За результатами проведених тестів можна зробити висновок, що розроблений алгоритм та програмний засіб значно удосконалюють якість відображення тривимірних об’єктів у доповненій реальності. Також можна зазначити, що у всіх тестах розроблений програмний засіб показав себе краще ніж usdz-файл.

Значним недоліком розробленого програмного засобу на відміну від програмного засобу для показу usdz-файлу є збільшена кількість кроків, яку користувач повинен зробити, щоб почати використовувати доповнену реальність. Також можна зазначити, що обидва програмних застосунків гублять позицію якорів при низькому освітленні. Слід звернути увагу на результат usdz-файлу при поганому освітленні. В цьому тесті видно велику перевагу розробленого способу генерації освітлення та тіней для об’єктів доповненої реальності.

Якщо дивитися на швидкодію, то можна сказати, що обидві програми впоралися чудово але є одне зауваження з приводу нагріву пристрою при використанні usdz-файлу. На момент використання розробленого програмного засобу температура апарату була значно менша ніж на момент використання вбудованого засобу перегляду usdz-файлів.

Було проведено експертне тестування розробленого програмного забезпечення. Згідно суб’єктивної оцінки експертів, розроблене програмне забезпечення працює на 33% краще.

РОЗДІЛ 5 СТАРТАП СКЛАДОВА ЧАСТИНА ПРОЕКТУ

5.1 Опис проблеми та дерево проблем

5.1.1 Анотація проекту

Проект спрямований на розробку та тестування способу та програмного забезпечення генерації освітлення та тіней для об’єктів у доповненій реальності. Розроблення передбачає винайдення способу генерації освітлення та тіней для об’єктів доповненої реальності з використанням комп’ютерного зору та різноманітних приладів мобільного пристрою. Тестування передбачає аналіз та обробку освітлення програмними засобами для подальшого використання розробленого способу в межах даного рішення.

5.1.2 Опис проблеми

На сьогоднішній день доповнена реальність набуває все більшої популярності. Провідні компанії, такі як Google та Apple, почали стрімко розвивати галузь безмаркерної доповненої реальності. Було випущено дві бібліотеки Google ARCore та Apple ARkit для мобільних пристроїв на базі Android a iOS відповідно. Ці бібліотеки надають можливість користувачам побачити тривимірні об’єкти у доповненій реальності без використання спеціальних маркерів. Провідні компанії-розробники почали використовувати доповнену реальність для демонстрації своїх продуктів користувачам. Отже, якість демонстрації є дуже важливим чинником для майбутньої популяризації технології доповненої реальності.

Розроблений спосіб являє собою покращений спосіб генерації освітлення та тіней для об’єктів у доповненій реальності, який дозволяє генерувати освітлення сцени доповненої реальності на основі освітлення обличчя з фронтальної камери. Програмне забезпечення є сценою доповненої реальності, що має під собою алгоритм генерації освітлення, за допомогою якого будуть створюватись освітлення та тіні. Метою даного

способу є покращення якості відображення об’єктів доповненої реальності, шляхом якісної генерації освітлення, а також ефективної обробки зображень.

Розроблені засоби дозволять збільшити якість відтворення доповненої реальності. Користувач зможе побачити більш якісне відображення доповненої реальності.

Основна спрямованість проекту – винайдення способу, що дозволить покращити відображення доповненої реальності.

5.1.3 Мета та завдання проекту відповідно до проблеми

Метою проекту є винайдення та тестування способу та програмного забезпечення генерації освітлення та тіней доповненої реальності. Для цього необхідно, виділити вже існуючі способи алгоритми та методи генерації освітлення та тіней доповненої реальності, виявити їх недоліки та переваги один над одними та обрати шлях розвитку проекту відповідно до виявлених переваг та недоліків. Проаналізувати предметну галузь та бажання користувачів, а також збільшити якість генерації освітлення та тіней доповненої реальності, прискорити обробку інформації, що дозволить надати проекту з винайдення способу подальшу його оптимізацію. Також ціль створити симулятор для отримання статистичної інформації для демонстрування та порівняння з аналогами винайденого способу. Відповідно презентувати отриманні результати для знаходження інвестицій для встановлення або переустановлення подібних систем чи заміни старіших і менш оптимальних. Залучити сторонніх розробників методом подальшого випуску вільного пакету програмних засобів, що дозволять розробникам створювати власні способи чи програмне забезпечення на основі розробленого способу, що дасть можливість використовувати та популізувати спосіб серед розробників.

5.1.4 Дерево проблем

Метою проекту є винайдення та тестування способу та програмного забезпечення генерації освітлення та тіней доповненої реальності. Для цього необхідно, виділити вже існуючі способи алгоритми та методи.

Дерево проблем проекту зображено на рис. 32. Як можна побачити з дерева проблем, неякісне відображення контенту у доповненій реальності, може призвести до небажаних наслідків.

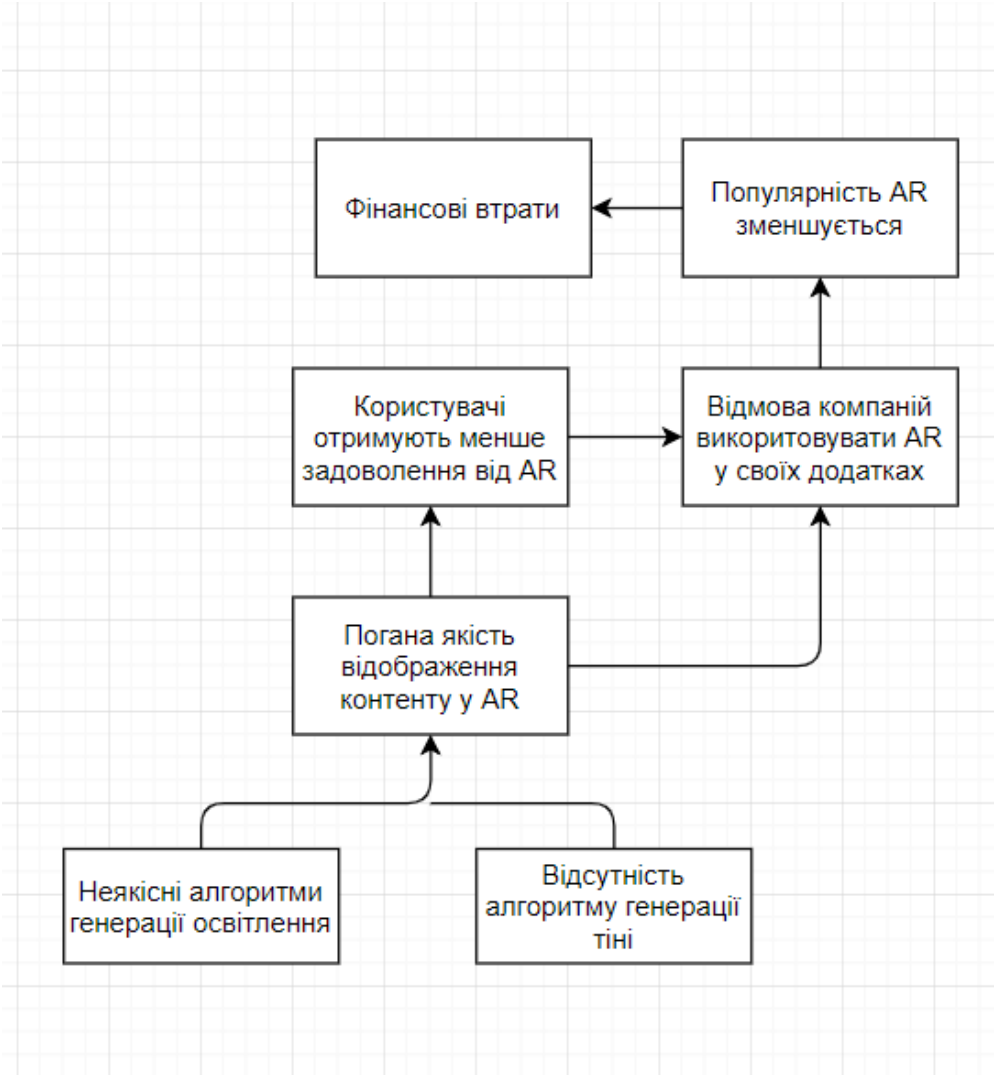


Рис. 32. Дерево проблем проекту

5.2 Аналіз зацікавлених сторін проекту

5.2.1 Зацікавлені сторони проекту

Кожен проект має свої зацікавлені сторони та оцінку їх важливостей. Зацікавлені сторони проекту та оцінка їх важливості та зацікавленості зібрані у таблиці 6 на сторінках 56 - 59.

Таблиця 6 – Зацікавлені сторони проекту

Група зацікавлених осіб	Інтереси групи в проекті	Умови довгого співробітництва з проектом	Важливість	Зацікавленість
Внутрішні зацікавлені сторони проекту				
Розробник проекту	Виконання проекту; Досягнення цільових показників проекту	Подальший розвиток проекту; Притік інвестицій	10	10
Команда керування проектом	Досягнення цільових показників проекту; Подальший розвиток технологій проекту	Подальший розвиток проекту; Притік інвестицій; Збільшення особистого доходу	10	10

Продовження таблиці 6

Інвестори проекту	Отримання зазначених доходів від участі в проекті; Подальший розвиток проекту; Досягнення цілей	Збільшення прибутку від інвестицій; Вигідніші умови підтримки проекту	10	9
Внутрішньо – корпоративні зацікавлені сторони проекту				
Менеджмент проекту	Розвиток проекту; Збереження притоку робочих місць;	Приріст робочих місць; Можливість кар’єрного росту; Реклама	7	4
Акціонери	Приріст доходності; Зростання загальної вартості проекту	Збільшення вартості проекту	10	9
Побічні співробітники	Кар’єрне зростання	Кар’єрний зростання	4	4

Продовження таблиці 6

Зовнішні зацікавлені сторони проекту				
Розробники ігор	Використання продукту проекту для покращення якості доповненої реальності; Якість власного продукту; Власний технологічний розвиток	Розвиток технологічного фону	5	6
Магазини	Використання продукту проекту для продажу власних товарів за допомогою доповненої реальності;	Розвиток проекту	8	9
Інші розробники програмного забезпечення	Використання продукту для покращення власного технологічного фону	Покращення старих технологій методами внесення змін у проект	7	8

Продовження таблиці 6

Побічні зацікавлені сторони проекту				
Розробники програмного забезпечення	Використання технологій доповненої реальності з ціллю покращення освітлення та тіней у власних додатках та збільшення доходу від додатка за рахунок впровадження розробленого способу	Оновлення технологій	7	6

5.2.2 Аналіз зацікавлених сторін проекту

Як видно з таблиці 5 розділу 5.2.1, зацікавлені сторони були розділені на 4 основні групи.

Внутрішні зацікавлені сторони проекту.

Тут зібрані особи, що напряму зацікавлені у проекті та його подальшому розвитку та встановленню на підприємствах закладах тощо, серед яких інвестори, для яких важливо мати прибуток з проекту, розробник проекту, який безпосередньо є керівником усього, що відбувається, як всередині команди так і зовні, команда проекту, що слідкує за виконанням проекту.

Внутрішньо-корпоративні зацікавлені сторони проекту

Ця група має менш серйозне відношення до проекту, проте без них його існування та подальший розвиток неможливі, серед них майбутні власники акцій, до яких входять і особи з першої групи, тому їх важливість та зацікавленість тісно перетинається з учасниками першої групи, а також до акціонерів належать побічні особи, зацікавленість яких може бути менша.

Побічні співробітники, їх важливість не дуже висока особливо на перших етапах розвитку, по-перше їх кількість невелика і по-друге вони не несуть такої сильної важливості, адже не пов'язані напряму з розробкою розроблюваного способу в певних випадках, проте якщо проект отримає подальший розвиток, їх кількість зросте, а співробітники, що були присутні на старті отримують більші доходи.

Зовнішні зацікавлені особи

Усі, хто зацікавлений у отриманні доступу до використання способу, тобто «цільова аудиторія» способу, до неї входять усі фірми, підприємства, корпорації, заклади, що працюють з доповненою реальністю або у сфері, яка потребує спосіб генерації освітлення та тіней у тому, чи іншому вигляді.

Побічні зацікавлені сторони

У супереч тому, що дана група має низьку зацікавленість та невисоку важливість, є потужним рушієм для популяризації способу, адже незалежні програмісти, можуть використовувати спосіб для створення власних продуктів у майбутньому, що може надати способу велику популярність, а також інвестиції у майбутньому.

5.3 Опис наукового проекту та технологій

Науковим проектом у рамках наукової дисертації є спосіб генерації тіні та освітлення для об'єктів у доповненій реальності.

Розвиток способів візуальної обробки інформації створює нові технології дослідження навколишнього світу та побудови інформаційної картини світу . У візуальному моделюванні розвиваються два напрями:

заміна реальності шляхом створення її віртуальної моделі в штучному інформаційному полі та доповнення реальності за рахунок створення багатошарової інформативної моделі, яка описуються людським уявленням. Перший напрямок називають віртуальним моделюванням, а модель, створена на її основі, називається віртуальною реальністю. Така назва підкреслює розрив між даною візуальною моделлю та реальним світом. Друга частина візуального моделювання базується на створенні складної моделі, в якій головною є реальність навколишнього світу, а візуальні моделі та моделювання доповнюють і уточнюють цю реальність. У цьому напрямку модель створюється не в штучному, а в реальному інформаційному полі. Цей напрямок створює модель, яка називається доповненою реальністю, підкреслюючи її зв'язок з реальним світом в альтернативу віртуальної реальності. Доповнена реальність або «розширена реальність» (augmented reality - AR) - результат введення в поле додаткових інформативних, сенсорних даних з метою розширення інформації про навколишній світ. Ця модель також називається "змішана реальність" (mixed reality - MR), створена з використанням комп'ютерного "доповнення" елементів прийнятної реальності. В області геоінформатики аналогом доповненої реальності є мультимасштабна карта, яка представляє собою умовну модель реальності, що зберігається в базі даних, у вигляді візуального відображення карт на різних масштабах і навіть окремих об'єктах. Мультимасштабна карта при необхідності робить суміш різноманітних карт різних масштабів для детального вивчення окремих фрагментів при збереженні загального представлення про відповідність різних фрагментів реальності, тобто при збереженні просторових відносин.

5.4 Бізнес рішення та основні характеристики бізнес – продукту

5.4.1 Резюме продукту

Результатом науково – дослідницької роботи буде програмне забезпечення та документація до нього, дане програмне забезпечення допоможе генерувати освітлення та тіні для об’єктів доповненої реальності. Далі буде розглянуто принцип розроблення, користь продукту, способи підтримки та подальший розвиток, що дасть абстрактне розуміння про продукт, що буде отриманий в результаті дослідницько роботи. Також далі будуть розглянуті можливості та шляхи вводу програмного забезпечення до систем різних типів, а також можливості отримання програмного забезпечення для подальшого використання у «домашніх умовах», також будуть описані необхідні вимоги для встановлення програмного забезпечення.

5.4.2 Опис продукту

Результатом дослідницької роботи буде програмне забезпечення але слід зазначити, що програмне забезпечення далеко не основний виробничий продукт, в основі програмного забезпечення лежить спосіб генерації освітлення та тіней, який можливо інтегрувати в інші рішення.

Основною перевагою при використанні способу є збільшення якості відображення тривимірних об’єктів у доповненій реальності, такий спосіб може знадобитися: іграм, магазинам з продажу різноманітних продукцій, додаткам, що працюють з доповненою реальністю.

Ігри зможуть надавати користувачеві більш якісний досвід від доповненої реальності.

Магазини з продажу різноманітних товарів зможуть показувати свої товари у якісній доповненій реальності.

Розробники програмного забезпечення зможуть інтегрувати даний спосіб в додатки пов’язані з доповненою реальністю.

Основною особливістю розроблення є простота в використанні та інтеграції в інші рішення. Спосіб буде реалізовано у вигляді додатку або бібліотеки, що підключається до іншого.

Процес «продажу» способу буде відбуватися методом інтегрування способу в інші проекти з використанням доповненої реальності та подальшою його підтримкою і оплатою за надані послуги щомісяця та за використання технології, у випадку з розробниками власних проектів їм буде надана можливість скористуватися демонстрацією способу, яку не вийде використати для своїх додатків. Для отримання доступу до технології необхідно замовити інтегрування в проект та обов'язково замовити подальшу підтримку, до якої буде прив'язаний окремий спеціаліст.

5.4.3 Конкурентні переваги

Дане рішення є інноваційним в наслідок того, що на поточний момент розробники не використовують такий детальний спосіб генерації освітлення та тіней для об'єктів у доповненій реальності. На даний момент існують рішення генерації освітлення та тіней у доповненій реальності але у всіх є свої значні недоліки, які погіршують відчуття користувача від доповненої реальності.

Таке рішення надасть користувачам більшу якість зображення при перегляді контенту у доповненій реальності, а розробникам дохід від продажу додатків з використанням даного способу.

Серед основних критеріїв у конкурентоздатності є:

1. Патентоздатність (новизна проекту);
2. Раціональність виробничої організаційної структури схеми;
3. Конкурентний персонал;
4. Прогресивність технології;
5. Прогресивність технологічних процесів та обладнання;
6. Науковий рівень розвитку розробників;
7. Науковий рівень системи;

8. Невимогливість до апаратного забезпечення;
9. Швидкодія.

Технологічні переваги способу(проекту):

1. Генерація освітлення для об’єктів доповненої реальності, згідно освітлення навколишнього середовища користувача, за рахунок використання розробленого алгоритму генерації освітлення.
2. Генерація тіней на основі згенерованого освітлення навколишнього середовища користувача.
3. Використання апаратних частин смартфону таких як датчик освітлення для покращення аналізу освітлення.
4. Використання алгоритму Light Estimation разом зі створеним способом.
5. Збільшення якості відображення тривимірних об’єктів у доповненій реальності за рахунок використання розробленого способу.
6. Реалізація у вигляді динамічної бібліотеки, що надасть змогу до імплементації способу до будь-якого іншого додатку з використанням доповненої реальності.
7. Зручний пакет розробника, що надасть додаткову мотивацію до використання способу «вільним» та побічним розробникам.
8. Кроссплатформеність, оскільки даний спосіб був реалізований на рушії Unity, то розроблений спосіб генерації освітлення та тіней для об’єктів доповненої реальності може бути підтриманий будь-якою платформою яку підтримує Unity. На поточний момент рушій Unity підтримує майже всі платформи.
9. Зручний відладчик на якому можна аналізувати роботу програмного продукту.
10. Зменшене навантаження на пристрій за рахунок оптимізації способу.

Загальні переваги проекту:

1. Так як проект та в майбутньому команда(фірма, компанія, тощо) не використовує підхід кредитування тому можна дозволити зручний та сталий формат формування цінової політики.
2. Весь персонал, що буде займатися обслуговуванням та оновленням програмного застосунку та налаштуванням у замовників, проходять постійні курси підготовки та розвитку, приймають участь у конференціях.
3. Швидка динаміка у оновленнях та підтримці ПО, відповідно до нових бібліотек доповненої реальності та нових пристроїв.
4. Відсутність некваліфікованих співробітників;
5. Встановлення та оновлення системи доповненої реальності відбувається за малий срок;
6. Онлайн тестування способу на прикладах, з можливістю встановити тестовий додаток з використання розробленого способу на свій пристрій.

Також слід приділити особливу увагу пункту про тестування, адже при впровадженні подібних способів на виробництво, замовник хоче бути впевненим у тому, що дана система буде давати необхідні результати, ця проблема вирішується методом реалізації окремої повноцінної, самостійної користувацької бібліотеки тестування, яка надасть змогу окрім симуляції роботи системи при різноманітних умовах та моніторингу, побудувати систему покращення роботи підприємства, а саме програмного забезпечення на яке вона встановлюється. Даний аспект надає суттєву перевагу, адже перед встановленням спосіб може бути повністю протестований, а результати покращення проаналізовані замовником.

Зменшене навантаження на пристрій за рахунок оптимізації (пункт 10 Технологічних переваг). На момент тестування створеного способу та програмного забезпечення для генерації освітлення та тіней було помічено набагато менший нагрів мобільного пристрою ніж у вмонтованому додатку у мобільного пристрою. Це є великою перевагою на користь використання

розробленого способу, адже нагрівання пристрою є вагомим аспектом у використанні мобільних додатків. За рахунок меншого нагрівання мобільного пристрою і зменшення навантаження, користувачі відчують кращий досвіт від роботи з технологіями доповненої реальності.

Основні переваги розробленого продукту:

1. Швидкодія;
2. Новизна;
3. Унікальність;
4. Модульність.

5.4.2 Клієнти та сегменти ринку

Як вже було зазначено у главі 5.2, аналіз зацікавлених сторін, основними клієнтами та споживачами даного способу є магазини з продажу різноманітних товарів та розробники програмного забезпечення з використанням технологій доповненої реальності. Детальна збірка інформації про клієнтів зібрана у таблиці 7.

Таблиця 7 – Клієнти та сегменти ринку

Тип закладу	Опис проблеми закладу, яку вирішує спосіб	Рейтинг споживача
Магазини з продажу різноманітних товарів	На сьогодні показ товарів у доповненій реальності є новим трендом до якого підлаштовуються магазини з різноманітним товаром. Доповнена реальність надає покупцям даних магазинів встановлювати додаток та переглядати товари у доповненій реальності. Це надає змогу покупцям побачити свій товар	10

Продовження таблиці 7

	перед покупкою, також надає можливість приміряти обраний товар. Наприклад виставити тривимірну модель дивана у доповненій реальності та подивитись чи підходить від до інтер'єру кімнати, чи ні. Найголовніше він надає покупцям так званий "УОУ ефект". Оскільки технологія доповненої реальності ще є досить молодою, то вона викликає у користувачів незабутні відчуття від процесу взаємодії. Цей "УОУ ефект" може гарно вплинути на загальну кількість продажу. Від чого магазин отримає значний прибуток.	
Розробники додатків з використанням доповненої реальності	Всі розробники намагаються якомога краще покращити свій продукт. Використовуючи розроблений спосіб, розробники програмного забезпечення зможуть збільшити якість свого програмного забезпечення, а також зменшити навантаження на апаратну частину своїх користувачів за рахунок	6

Продовження таблиці 7

	оптимізованого способу генерації освітлення та тіней для об’єктів доповненої реальності	
Розробники бібліотек доповненої реальності	Для розробників бібліотек доповненої реальності дуже важливо бути конкурентоздатними. На поточний момент цими розробниками є такі гіганти як Google та Apple. На поточний момент вони розробили бібліотеки ARkit та ARCore. Які є конкурентами у сфері доповненої реальності. Конкуренція у технологія може бути порівняна з можливістю операційних систем в цілому.	8

Як видно з таблиці, що наведено вище основними клієнтами є магазини з продажу різноманітних товарів, розробники програмного забезпечення з використанням технологій доповненої реальності та розробники бібліотек для взаємодії з доповненою реальністю. Цей сегмент має великий об’єм інвестицій. Поле «Рейтинг споживачів» таблиці 3, вказує на сумарний коефіцієнт на необхідність конкретному типу споживачів у відношенні до того чи готові вони впроваджувати подібні технології та чи потрібні вони їм взагалі, наприклад для розробників програмного забезпечення з використанням доповненої реальності метод впровадження способу піде на користь, через прискорення роботи та збільшення якості перегляду тривимірних об’єктів у доповненій реальності, проте готовність

встановлювати та оновлювати подібне програмне забезпечення менша ніж у магазинах з продажу різноманітних товарів.

5.5 Унікальна цінність способу

Як вже згадувалося раніше існує декілька способів генерації освітлення та тіней для об’єктів доповненої реальності, але вони мають свої недоліки. В основі способу лежить алгоритм генерації освітлення, який аналізує освітлення простору користувача і потім генерує освітлення основуючись на проаналізованих даних. Це і є так званою новизною рішення, що дозволить збільшити якість показу тривимірних об’єктів у доповненій реальності.

На поточний момент можна виділити наступний унікальний функціональність способу:

1. Спосіб та програмний засіб надає змогу генерувати реалістичне освітлення ґрунтуючись на даних отриманих з обличчя користувача. На поточний момент є алгоритми та способи, які надають таку змогу але жоден с них не використовує отримані дані для генерації освітлення саме у доповненій реальності.
2. Спосіб та програмний засіб надає змогу генерувати якісну тінь для об’єктів доповненої реальності ґрунтуючись на отриманих даних освітлення. Таким чином користувач може отримати краще відчуття від користування доповненою реальністю.
3. Кроссплатформеність. На поточний момент спосіб може бути впроваджений у будь-яку бібліотеку для взаємодії з доповненою реальність. Тобто крім мобільної доповненої реальності, спосіб можуть використовувати такі пристрої як Microsoft Hololens, окуляри доповненої реальності Magic Leap, Google Glass, окуляри Vuzix Blad AR, Meta 2, Optiment Ora, Solos. Це все можливо завдяки розробці на кроссплатформеному рушії Unity.

5.6 Доходи і витрати

5.6.1 Витрати

Для прийняття рішення щодо інвестиційного проекту всі витрати, пов'язані з його здійсненням, необхідно розподілити на інвестиційні та виробничі.

Загальна сума витрат на здійснення проекту включає:

1. Витрати на формування основного капіталу містять початкові і поточні інвестиції;
2. Витрати на формування оборотного капіталу;
3. Виробничі витрати.

Всі інвестиційні потреби підприємства можна поділити на три групи:

1. Прямі інвестиції – це безпосередньо необхідні для реалізації інвестиційного проекту;
2. Супутні інвестиції – це вкладення в об'єкти, безпосередньо технологічно які пов'язані із забезпеченням нормальної експлуатації;
3. Інвестування виконання науково-дослідних робіт.

До складу початкових інвестицій відносяться:

1. Витрати на передінвестиційні дослідження, проведення дослідницьких і конструкторських робіт, на розробку проектних матеріалів, на робоче проектування і прив'язку проекту;
2. Витрати на придбання та оренду потужностей, включаючи вартість підготовки до освоєння;
3. Витрати на придбання та доставку машин, обладнання, інструменту та інвентарю, в тому числі імпортних;
4. Витрати на приймально-здавальні випробування;

5. Витрати на пусконалагоджувальні роботи, комплексне освоєння проектних потужностей і досягнення проектних техніко-економічних показників;
6. Витрати на придбання патентів, ліцензій, ноу-хау, технологій та інших амортизаційних нематеріальних активів;
7. Витрати на підготовку кадрів для об'єктів, що вводяться в дію;
8. Одноразові виплати, зокрема страховим організаціям;
9. Витрати, що виникають при створенні та реєстрації фірми (оплата юридичних послуг зі складання установчих документів, витрати на реєстрацію фірми і оформлення прав власності;
10. Витрати на підготовчі дослідження;
11. Витрати, пов'язані з діяльністю персоналу в період підготовки виробництва (оплата праці, витрати, утримання приміщень, автомобілів, комп'ютерів та іншого обладнання), не враховані в кошторисної вартості об'єкт.

Висновки по витратам:

Вартість проекту = ціна проектування + ціна розроблення + активна підтримка і доопрацювання після впровадження + оплата за компоненти та сервіси + оплата за рекламу і просування. На рис. 33 схематично зображено 2 етапи витрат.

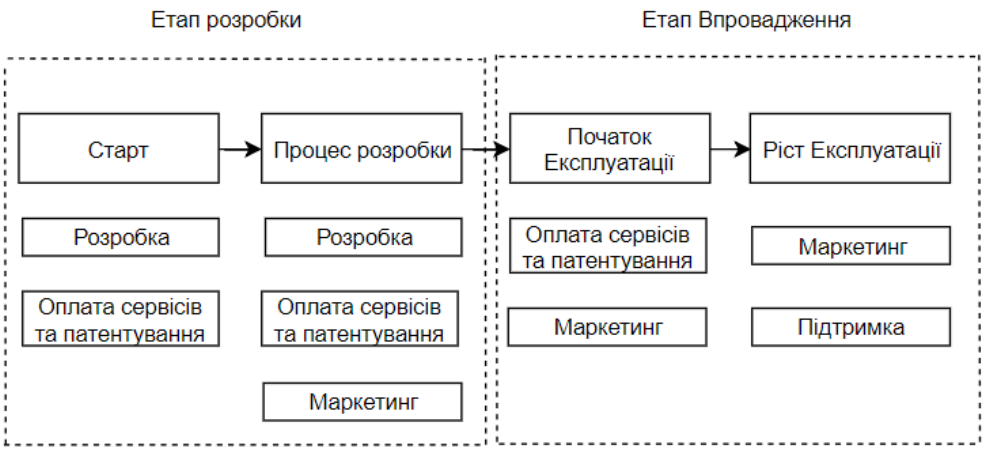


Рис. 33. Види витрат на проект на різних етапах.

5.6.2 Доходи

Як зрозуміло з попередніх пунктів та ринку споживачів, товаром є спосіб для генерації освітлення та тіней об’єктів доповненої реальності. Основними джерелом доходу будуть кошти з експлуатації даного способу. Загальну дохідність проекту в майбутньому буде прийнято рахувати метриками доходності, такими, як:

1. Bookings (замовлення)**;**
2. Revenue (дохід);
3. TCV –Total Contract Value (загальна вартість контракту);
4. LTV – Life Time Value (тотальна цінність);
5. Unearned or Deferred Revenue (майбутні доходи).

Bookings – це оцінка вартості контракту між компанією і клієнтом. Ця метрика відбиває зобов'язання клієнта заплатити компанії зазначені в договорі гроші. Так як основою експлуатації та впровадження на виробництво методу є контракт дана метрика, грає основну роль в обрахунках доходності проекту, сумарна вартість контракту на один заклад.

Про прибуток (*Revenue*) можна говорити тоді, коли послуга вже надана або буде надаватися регулярно протягом зазначеного в договорі терміну підписки. Є вихідним фактором першої метрики, рахується по закінченню контракту, тобто при вдалому завершенні.

TCV – це загальна вартість контракту, ціна одиниці виміру товару та загальна вартість товару або вартість виконаних робіт (наданих послуг), що поставляються згідно з договором, крім випадків, коли ціна товару розраховується за формулою, а також валюта контракту. Якщо згідно з договором постачаються товари різної якості та асортименту, ціна встановлюється окремо за одиницю товару кожного сорту, марки, а окремим пунктом договору зазначається його загальна вартість. У цьому разі цінові показники можуть бути зазначені в додатках (специфікаціях), на які робиться посилання в тексті договору.

LTV - це передбачення чистого доходу, пов'язаного з усіма майбутніми відносинами з клієнтом. Модель прогнозу може мати різні рівні точності, що варіюються від приблизних, евристичних, до складних, що використовують техніки предикативного аналізу. Довічну цінність клієнта також можна описати як грошову вартість відносин з клієнтом на підставі поточних розмірів видимих майбутніх грошових потоків від відносин з клієнтом. Пожиттєва цінність клієнта – це важливе поняття в тому плані, що воно стимулює перенесення фокуса компаній з квартальних доходів на довгострокові здорові відносини зі своїми клієнтами. Пожиттєва цінність клієнта – це важливий показник, оскільки вона являє собою верхній поріг витрат на покупку нових клієнтів. Одна з перших згадок цього терміну трапилася 1988 року в книзі Database Marketing, що містить детальні робочі приклади.

UDR - Відстрочені доходи (також відомі як неотримані доходи або неотриманий прибуток), нараховується як грошові кошти отриманні за товари чи послуги, які ще не були доставлені. Відповідно до принципу характеристики доходу, він відображається як відстрочений до моменту здійснення доставки, в цей час він перетворюється у дохід.

Типовим прикладом є річний договір про технічне обслуговування, на якому весь контракт виставляється на рахунку. "Я отримав 12000 доларів за річний контракт на технічне обслуговування, але повинен визнати його як дохід від майбутніх доходів, а потім заробляти 1000 доларів щомісяця за надання послуг".

5.6.3 Таблиця з витратами та доходами

Таблиця 8 описує стадії розвитку проекту, розділи витрат та сумарний дохід. Слід врахувати, що перший та другий етапи є разовими і є допроектими, третій етап розраховується помісячно, після закінчення перших двох етапів.

Таблиця 8 – Стадії розвитку проекту

Етап проекту	Статті витрат	Шляхи доходу	Витрати	Дохід	Сума
Етап розроблення	Розроблення, оплата сервісів, патентування, маркетинг		500 у.о	0	-500у.о
Етап впровадження	Маркетинг, підтримка, реалізація, впровадження	Впровадження у тестовому режимі на виробництво	1.000 у.о	0 у.о	-1500 у.о
Етап експлуатації	Підтримка, обслуговування, розвиток проекту, оплата персоналу	Замовлення, оплата послуг замовниками, підтримка, підписка, оплата тестового додатку, отримання відсотку з продажу товарів, впровадження способу в інші додатки, створення власного продукту реалізуючого спосіб.	3000 у.о	40.000 у.о	35.500у.о

5.7 Бізнес - модель

На рис. 34 зображена загальна схема бізнес моделі.

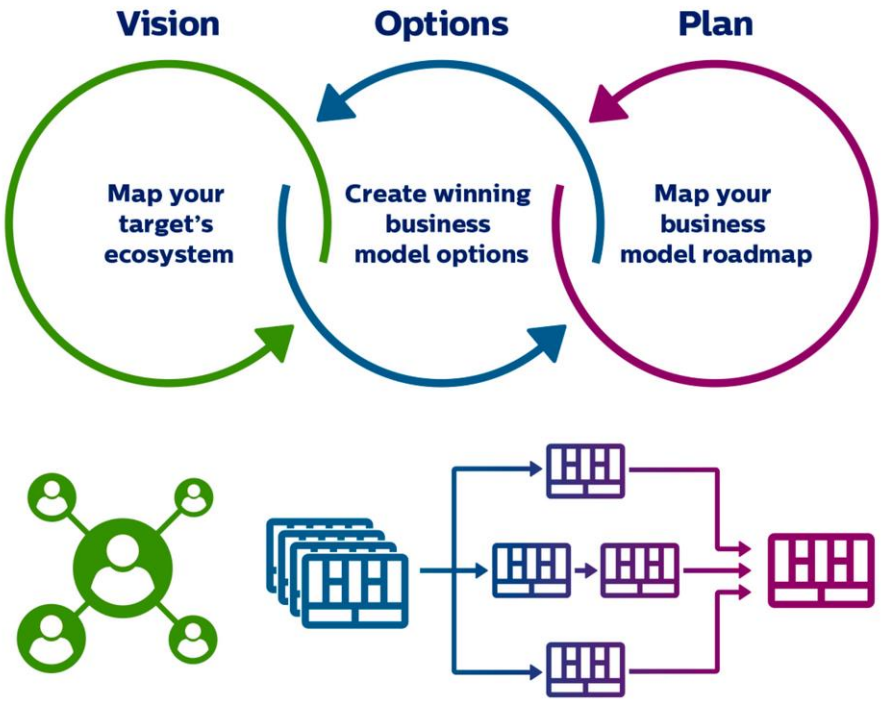


Рис. 34. Загальна бізнес модель

Сегмент користувачів:

Для кого? – компанії, які хочуть або вже використовують технології доповненої реальності.

Найважливіші клієнти? – Магазини з різноманітними товарами, котрі хочуть використовувати доповнену реальність для збільшення свого доходу.

Пропозиція:

Яку цінність від способу отримують клієнти? – Клієнти отримують збільшення якості відображення тривимірних об'єктів у доповненій реальності.

Яку проблему користувачів ми вирішуємо? – Погану якість відображення об'єктів доповненої реальності.

Яку необхідність користувачів ми задовольняємо? – Користувачам потрібно максимальне занурення у технології доповненої реальності.

Які набори продуктів ми пропонуємо? – Спосіб та програмне забезпечення для генерації освітлення та тіней для тривимірних об’єктів доповненої реальності.

Канали розповсюдження:

Через які канали ми хочемо досягти кожен сегмент наших споживачів? – реклама та створення вільної бібліотеки, розповсюдження її у всіх можливих пакетах для різних мов програмування, Unity Asset Store.

Який з каналів розповсюдження є найпривабливішим? – можливість розповсюдження методу, через вільні бібліотеки програмного коду.

Потоки доходу:

За що готові платити користувачі? – за використання способу для генерації освітлення та тіней об’єктів доповненої реальності для збільшення якості відображення доповненої реальності.

Яким чином вони можуть платити? – так, як основними клієнтами є магазини різноманітних товарів, то оплата послуг буде укладатися за контрактами в яких буде описано умови підтримки, тестування та оновлення, контракт має можливість продовжуватись, або є можливість заключити його як підписку на певний статичний час.

Альтернативні шляхи? – дохід також може бути нарахований як відсоток від проданих товарів за допомогою використання доповненої реальності користувачами.

Основні ресурси :

Необхідні ресурси? – набір різноманітних девайсів для тестування.

Шляхи знаходження необхідних ресурсів? – безкоштовно тестові пакети для закладів, від чого виграє проект і заклади.

Відносини з клієнтами? – замовник підписуючи один з видів контракту, отримує набір послуг, окрім впровадження способу, це може бути обслуговування, оновлення, налаштування або консультація.

Потоки доходів? – основний дохід йде від магазинів з продажу різноманітних товарів, які хочуть використовувати новітні технології

доповненої реальності для збільшення прибутку з продажу власних товарів, а також невелика частина прибуток з компаній, які займаються розробкою у сфері доповненої реальності. Підприємства підписують контракт за певним пакетом послуг, який коштує по різному і містить різні набори послуг окрім безпосереднього впровадження способу.

5.8 Висновки до розділу

Було описано проблеми проекту та створено дерево проблем проекту. Було проаналізовано зацікавлені сторони проекту, описано науковий проект та технології проекту. Охарактеризовано бізнес рішення проекту та наведені основні характеристики бізнес-продукту. Було сформульована унікальна цінність способу генерації освітлення та тіней об’єктів доповненої реальності. Були підраховані доходи і витрати проекту. Була сконструйована бізнес-модель.

ВИСНОВКИ

За час створення магістерської дисертації були проаналізовані існуючі рішення у сфері доповненої реальності та алгоритми генерації освітлення та тіней. Були наведені математичні засади для реалізації способу генерації освітлення та тіней об’єктів доповненої реальності. Згідно яких, було детально обґрунтовано створення модифікованого способу генерації освітлення та тіней об’єктів доповненої реальності. Сформульовано методику оцінювання способу генерації освітлення та тіней, згідно якої було протестовано розроблену програмну реалізацію способу. Також було зібрано суб’єктивну експертну оцінку, яка показала, що розроблений програмний засіб працює на 33% краще за еталон. На основі сформульованої методики, було проведено об’єктивне тестування розробленого програмного застосунку, яке показало, що середнє значення удосконалення оцінки показника температури пристрою, створеного програмного забезпечення, становить 15%, що є дуже високим показником. На основі створеного способу генерації освітлення було сформульовано стартап складову частину проекту. За час розробки проекту було сформульовано опис проблеми та побудовано дерево проблем, проаналізовано зацікавлені сторони проекту, описано науковий проект та технології проекту, сформульовані бізнес рішення та основні характеристики бізнес-продукту, охарактеризована унікальна цінність способу, підраховані доходи і витрати проекту та побудована бізнес модель

СПИСОК ВИКОРИСТАНИХ ЛІТЕРАТУРНИХ ДЖЕРЕЛ

1. Buerli M., Misslinger S. Introducing ARKit-Augmented Reality for iOS //Apple Worldwide Developers Conference (WWDC'17). – 2017. – С. 1-187.
2. Gao L. et al. Real-time visual representations for mobile mixed reality remote collaboration //SIGGRAPH Asia 2018 Virtual & Augmented Reality. – ACM, 2018. – С. 15.
3. Nugroho A., Pramono B. A. Aplikasi Mobile Augmented Reality Berbasis Vuforia Dan Unity Pada Pengenalan Objek 3D Dengan Studi Kasus Gedung M Universitas Semarang //Jurnal Transformatika. – 2017. – Т. 14. – №. 2. – С. 86-91.
4. Evans G. et al. Evaluating the Microsoft HoloLens through an augmented reality assembly application //Degraded Environments: Sensing, Processing, and Display 2017. – International Society for Optics and Photonics, 2017. – Т. 10197. – С. 101970V.
5. Yael S. Pyramid-Based Shadow Removal. – 2017.
6. Vidal A. R. et al. Ultimate SLAM? Combining Events, Images, and IMU for Robust Visual SLAM in HDR and High-Speed Scenarios //IEEE Robotics and Automation Letters. – 2018. – Т. 3. – №. 2. – С. 994-1001.
7. Debevec P. A median cut algorithm for light probe sampling //ACM Siggraph 2005 Posters. – ACM, 2005. – С. 66.
8. Kautz J., Snyder J., Sloan P. P. J. Fast Arbitrary BRDF Shading for Low-Frequency Lighting Using Spherical Harmonics //Rendering Techniques. – 2002. – Т. 2. – №. 291-296. – С. 1.
9. Indraprastha A., Shinozaki M. The investigation on using Unity3D game engine in urban design study //Journal of ICT Research and Applications. – 2009. – Т. 3. – №. 1. – С. 1-18.
10. Hejlsberg A., Wiltamuth S., Golde P. C# language specification. – Addison-Wesley Longman Publishing Co., Inc., 2003.

11. Allan A. Learning iPhone Programming: From Xcode to App Store. – " O'Reilly Media, Inc.", 2010.
12. Randolph N. et al. Professional visual studio 2010. – John Wiley & Sons, 2010.
13. Li P. Selecting and using virtualization solutions: our experiences with VMware and VirtualBox //Journal of Computing Sciences in Colleges. – 2010. – Т. 25. – №. 3. – С. 11-17.

ДОДАТКИ

Додаток 1

Лістинг програмного коду

```
public class UnityARKitControl : MonoBehaviour {

        UnityARSessionRunOption [] runOptions = new UnityARSessionRunOption[4];
        UnityARAlignment [] alignmentOptions = new UnityARAlignment[3];
        UnityARPlaneDetection [] planeOptions = new UnityARPlaneDetection[4];

        int currentOptionIndex = 0;
        int currentAlignmentIndex = 0;
        int currentPlaneIndex = 0;

    // Use this for initialization
    void Start () {
            runOptions [0] =
UnityARSessionRunOption.ARSessionRunOptionRemoveExistingAnchors |
UnityARSessionRunOption.ARSessionRunOptionResetTracking;
            runOptions [1] = UnityARSessionRunOption.ARSessionRunOptionResetTracking;
            runOptions [2] =
UnityARSessionRunOption.ARSessionRunOptionRemoveExistingAnchors;
            runOptions [3] = 0;

            alignmentOptions [0] = UnityARAlignment.UnityARAlignmentCamera;
            alignmentOptions [1] = UnityARAlignment.UnityARAlignmentGravity;
            alignmentOptions [2] = UnityARAlignment.UnityARAlignmentGravityAndHeading;

            planeOptions [0] = UnityARPlaneDetection.Horizontal;
            planeOptions [1] = UnityARPlaneDetection.None;

    }

    // Update is called once per frame
    void Update () {

    }

        void OnGUI()
        {
            if (GUI.Button (new Rect (100, 100, 200, 50), "Stop")) {
                UnityARSessionNativeInterface.GetARSessionNativeInterface ().Pause ();
            }

            if (GUI.Button (new Rect (300, 100, 200, 50), "Start")) {
                ARKitWorldTrackingSessionConfiguration sessionConfig = new
ARKitWorldTrackingSessionConfiguration (alignmentOptions [currentAlignmentIndex],
planeOptions[currentPlaneIndex]);
                UnityARSessionNativeInterface.GetARSessionNativeInterface
().RunWithConfigAndOptions (sessionConfig, runOptions[currentOptionIndex]);
            }


            if (GUI.Button (new Rect (100, 300, 200, 100), "Start Non-WT Session")) {
                ARKitSessionConfiguration sessionConfig = new
ARKitSessionConfiguration (alignmentOptions [currentAlignmentIndex], true, true);
                UnityARSessionNativeInterface.GetARSessionNativeInterface
().RunWithConfig (sessionConfig);
            }


            string runOptionStr = (currentOptionIndex == 0 ? "Full" :
(currentOptionIndex == 1 ? "Tracking" : (currentOptionIndex == 2 ? "Anchors" :
"None")));
```

```
            if (GUI.Button (new Rect (100, 200, 150, 50), "RunOption:" +
runOptionStr)) {
                currentOptionIndex = (currentOptionIndex + 1) % 4;
            }

            string alignmentOptionStr = (currentAlignmentIndex == 0 ? "Camera" :
(currentAlignmentIndex == 1 ? "Gravity" :  "GravityAndHeading"));
            if (GUI.Button (new Rect (300, 200, 150, 50), "AlignOption:" +
alignmentOptionStr)) {
                currentAlignmentIndex = (currentAlignmentIndex + 1) % 3;
            }

              string planeOptionStr = currentPlaneIndex == 0 ? "Horizontal":  "None";
            if (GUI.Button (new Rect (500, 200, 150, 50), "PlaneOption:" +
planeOptionStr)) {
                currentPlaneIndex = (currentPlaneIndex + 1) % 2;
            }
        }
    }


using System.Collections;
using System.Collections.Generic;
using UnityEngine;
using UnityEngine.XR.iOS;

public class UnityARCameraManager : MonoBehaviour {

    public Camera m_camera;
    private UnityARSessionNativeInterface m_session;
    private Material savedClearMaterial;

    [Header("AR Config Options")]
    public UnityARAlignment startAlignment = UnityARAlignment.UnityARAlignmentGravity;
    public UnityARPlaneDetection planeDetection = UnityARPlaneDetection.Horizontal;
    public bool getPointCloud = true;
    public bool enableLightEstimation = true;
    public bool enableAutoFocus = true;
     public UnityAREnvironmentTexturing environmentTexturing =
UnityAREnvironmentTexturing.UnityAREnvironmentTexturingNone;

    [Header("Image Tracking")]
    public ARReferenceImagesSet detectionImages = null;
    public int maximumNumberOfTrackedImages = 0;

    [Header("Object Tracking")]
    public ARReferenceObjectsSetAsset detectionObjects = null;
    private bool sessionStarted = false;

    public ARKitWorldTrackingSessionConfiguration sessionConfiguration
    {
        get
        {
            ARKitWorldTrackingSessionConfiguration config = new
ARKitWorldTrackingSessionConfiguration ();
            config.planeDetection = planeDetection;
            config.alignment = startAlignment;
            config.getPointCloudData = getPointCloud;
            config.enableLightEstimation = enableLightEstimation;
            config.enableAutoFocus = enableAutoFocus;
            config.maximumNumberOfTrackedImages = maximumNumberOfTrackedImages;
            config.environmentTexturing = environmentTexturing;
            if (detectionImages != null)
                config.referenceImagesGroupName = detectionImages.resourceGroupName;
```

```
            if (detectionObjects != null)
            {
                config.referenceObjectsGroupName = "";
    config.dynamicReferenceObjectsPtr =
m_session.CreateNativeReferenceObjectsSet(detectionObjects.LoadReferenceObjectsInSet()
);
            }

            return config;
        }
    }

    // Use this for initialization
    void Start () {

        m_session = UnityARSessionNativeInterface.GetARSessionNativeInterface();

        Application.targetFrameRate = 60;

        var config = sessionConfiguration;
        if (config.IsSupported) {
            m_session.RunWithConfig (config);
            UnityARSessionNativeInterface.ARFrameUpdatedEvent += FirstFrameUpdate;
        }

        if (m_camera == null) {
            m_camera = Camera.main;
        }
    }

    void OnDestroy()
    {
        m_session.Pause();
    }

    void FirstFrameUpdate(UnityARCamera cam)
    {
        sessionStarted = true;
        UnityARSessionNativeInterface.ARFrameUpdatedEvent -= FirstFrameUpdate;
    }

    public void SetCamera(Camera newCamera)
    {
        if (m_camera != null) {
            UnityARVideo oldARVideo = m_camera.gameObject.GetComponent<UnityARVideo>
();
            if (oldARVideo != null) {
                savedClearMaterial = oldARVideo.m_ClearMaterial;
                Destroy (oldARVideo);
            }
        }
        SetupNewCamera (newCamera);
    }

    private void SetupNewCamera(Camera newCamera)
    {
        m_camera = newCamera;

        if (m_camera != null) {
            UnityARVideo unityARVideo = m_camera.gameObject.GetComponent<UnityARVideo>
();
            if (unityARVideo != null) {
                savedClearMaterial = unityARVideo.m_ClearMaterial;
                Destroy (unityARVideo);
            }
```

```
            unityARVideo = m_camera.gameObject.AddComponent<UnityARVideo> ();
            unityARVideo.m_ClearMaterial = savedClearMaterial;
        }
    }

    // Update is called once per frame

    void Update () {

        if (m_camera != null && sessionStarted)
        {
            Matrix4x4 matrix = m_session.GetCameraPose();
            m_camera.transform.localPosition = UnityARMatrixOps.GetPosition(matrix);
            m_camera.transform.localRotation = UnityARMatrixOps.GetRotation (matrix);

            m_camera.projectionMatrix = m_session.GetCameraProjection ();
        }

    }

}

public class UnityARGeneratePlane : MonoBehaviour
    {
         public GameObject planePrefab;
        private UnityARAnchorManager unityARAnchorManager;

         // Use this for initialization
         void Start () {
            unityARAnchorManager = new UnityARAnchorManager();
              UnityARUtility.InitializePlanePrefab (planePrefab);
         }

        void OnDestroy()
        {
            unityARAnchorManager.Destroy ();
        }

        void OnGUI()
        {
              IEnumerable<ARPlaneAnchorGameObject> arpags =
unityARAnchorManager.GetCurrentPlaneAnchors ();
              foreach(var planeAnchor in arpags)
              {
                //ARPlaneAnchor ap = planeAnchor;
                //GUI.Box (new Rect (100, 100, 800, 60), string.Format ("Center:
x:{0}, y:{1}, z:{2}", ap.center.x, ap.center.y, ap.center.z));
                //GUI.Box(new Rect(100, 200, 800, 60), string.Format ("Extent: x:{0},
y:{1}, z:{2}", ap.extent.x, ap.extent.y, ap.extent.z));
              }
        }
    }
```

Лістинг шейдеру для отримання тіні

```
Shader "Shadow/TransparentShadowReceiver"
{

     Properties
     {
          // Usual stuffs
          _Color("Main Color", Color) = (1,1,1,1)
          _SpecColor("Specular Color", Color) = (0.5, 0.5, 0.5, 0)
          _Shininess("Shininess", Range(0.01, 1)) = 0.078125
          _MainTex("Base (RGB) TransGloss (A)", 2D) = "white" {}

     // Bump stuffs
     //_Parallax ("Height", Range (0.005, 0.08)) = 0.02
     _BumpMap("Normalmap", 2D) = "bump" {}
     //_ParallaxMap ("Heightmap (A)", 2D) = "black" {}

     // Shadow Stuff
     _ShadowIntensity("Shadow Intensity", Range(0, 1)) = 0.6
     }


          SubShader
     {
          Tags {
          "Queue" = "AlphaTest"
          "IgnoreProjector" = "True"
          "RenderType" = "Transparent"
          }

          LOD 300


          // Main Surface Pass (Handles Spot/Point lights)
          CGPROGRAM
                    #pragma surface surf BlinnPhong alpha vertex:vert fullforwardshadows
approxview

                    half _Shininess;

                    sampler2D _MainTex;
                    float4 _Color;
                    sampler2D _BumpMap;
                    //sampler2D _ParallaxMap;
                    float _Parallax;

                    struct v2f {
                         V2F_SHADOW_CASTER;
                         float2 uv : TEXCOORD1;
                    };

                    struct Input {
                         float2 uv_MainTex;
                         float2 uv_BumpMap;
                         //float3 viewDir;
                    };

                    v2f vert(inout appdata_full v) {
                         v2f o;
                         return o;
                    }

                    void surf(Input IN, inout SurfaceOutput o) {
```

```
                        // Comment the next 4 following lines to get a standard bumped
rendering
                        // [Without Parallax usage, which can cause strange result on
the back side of the plane]
                        /*half h = tex2D (_ParallaxMap, IN.uv_BumpMap).w;
                        float2 offset = ParallaxOffset (h, _Parallax, IN.viewDir);
                        IN.uv_MainTex += offset;
                        IN.uv_BumpMap += offset;*/

                        fixed4 tex = tex2D(_MainTex, IN.uv_MainTex);
                        o.Albedo = tex.rgb * _Color.rgb;
                        o.Gloss = tex.a;
                        o.Alpha = tex.a * _Color.a;
                        //clip(o.Alpha - _Cutoff);
                        o.Specular = _Shininess;
                        o.Normal = UnpackNormal(tex2D(_BumpMap, IN.uv_BumpMap));
                   }
          ENDCG



               // Shadow Pass : Adding the shadows (from Directional Light)
               // by blending the light attenuation
               Pass {
                    Blend SrcAlpha OneMinusSrcAlpha
                    Name "ShadowPass"
                    Tags {"LightMode" = "ForwardBase"}

                    CGPROGRAM
// Upgrade NOTE: excluded shader from DX11; has structs without semantics (struct v2f
members lightDir)
#pragma exclude_renderers d3d11
                    #pragma vertex vert
                    #pragma fragment frag
                    #pragma multi_compile_fwdbase
                    #pragma fragmentoption ARB_fog_exp2
                    #pragma fragmentoption ARB_precision_hint_fastest
                    #include "UnityCG.cginc"
                    #include "AutoLight.cginc"

                    struct v2f {
                        float2 uv_MainTex : TEXCOORD1;
                        float4 pos : SV_POSITION;
                        LIGHTING_COORDS(3,4)
                        float3   lightDir;
                    };

                    float4 _MainTex_ST;

                    sampler2D _MainTex;
                    float4 _Color;
                    float _ShadowIntensity;

                    v2f vert(appdata_full v)
                    {
                        v2f o;
                        o.uv_MainTex = TRANSFORM_TEX(v.texcoord, _MainTex);
                        o.pos = UnityObjectToClipPos(v.vertex);
                        o.lightDir = ObjSpaceLightDir(v.vertex);
                        TRANSFER_VERTEX_TO_FRAGMENT(o);
                        return o;
                    }

                    float4 frag(v2f i) : COLOR
                    {
```

```
                    float atten = LIGHT_ATTENUATION(i);

                    half4 c;
                    c.rgb = 0;
                    c.a = (1 - atten) * _ShadowIntensity * (tex2D(_MainTex,
i.uv_MainTex).a);
                    return c;
                }
                ENDCG
            }


    }

        FallBack "Transparent/Cutout/VertexLit"
}

Light [] lightsInScene;
    SphericalHarmonicsL2 shl;
    public static SphericalHarmonicsL2 tmp;
    // Use this for initialization
    void Start () {
        lightsInScene = FindAllLights();
        shl = new SphericalHarmonicsL2 ();
        UnityARSessionNativeInterface.ARFrameUpdatedEvent += UpdateLightEstimations;

    }

    void OnDestroy()
    {
        UnityARSessionNativeInterface.ARFrameUpdatedEvent -= UpdateLightEstimations;
    }


    Light []  FindAllLights()
    {
        return FindObjectsOfType<Light> ();
    }



    void UpdateLightEstimations(UnityARCamera camera)
    {
        if (camera.lightData.arLightingType == LightDataType.LightEstimate) {
            UpdateBasicLightEstimation (camera.lightData.arLightEstimate);
        }
        else if (camera.lightData.arLightingType ==
LightDataType.DirectionalLightEstimate)
        {
            UpdateDirectionalLightEstimation
(camera.lightData.arDirectonalLightEstimate);
        }
    }

    void UpdateBasicLightEstimation(UnityARLightEstimate uarle)
    {
        foreach (Light l in lightsInScene)
        {
            float newai = uarle.ambientIntensity;
            l.intensity = newai / 1000.0f;
            l.colorTemperature = uarle.ambientColorTemperature;
        }
```

```
    }

    void UpdateDirectionalLightEstimation(UnityARDirectionalLightEstimate uardle)
    {
         for (int colorChannel = 0; colorChannel < 3; colorChannel++) {
              for (int index = 0; index < 9; index++) {
                   shl [colorChannel, index] = uardle.sphericalHarmonicsCoefficients
[(colorChannel * 9) + index];
              }
         }

         if (LightmapSettings.lightProbes != null) {
              int probeCount = LightmapSettings.lightProbes.count;

              if (probeCount > 0) {

                   SphericalHarmonicsL2[] bakedProbes =
LightmapSettings.lightProbes.bakedProbes;

                   for (int i = 0; i < probeCount; i++) {
                        bakedProbes [i] = shl;
                   }
              }
         }

         RenderSettings.ambientProbe = shl;
         RenderSettings.ambientMode = AmbientMode.Custom;
        tmp = shl;

    }

public class UnityARHitTestExample : MonoBehaviour
    {
         public Transform m_HitTransform;
         public float maxRayDistance = 30.0f;
         public LayerMask collisionLayer = 1 << 10;  //ARKitPlane layer

        bool HitTestWithResultType (ARPoint point, ARHitTestResultType resultTypes)
        {
            List<ARHitTestResult> hitResults =
UnityARSessionNativeInterface.GetARSessionNativeInterface ().HitTest (point,
resultTypes);
            if (hitResults.Count > 0) {
                foreach (var hitResult in hitResults) {
                    Debug.Log ("Got hit!");
                    m_HitTransform.position = UnityARMatrixOps.GetPosition
(hitResult.worldTransform);
                    m_HitTransform.rotation = UnityARMatrixOps.GetRotation
(hitResult.worldTransform);
                    Debug.Log (string.Format ("x:{0:0.######} y:{1:0.######}
z:{2:0.######}", m_HitTransform.position.x, m_HitTransform.position.y,
m_HitTransform.position.z));
                    return true;
                }
            }
            return false;
        }


         void Update () {
              #if UNITY_EDITOR           if (Input.GetMouseButtonDown (0)) {
                   Ray ray = Camera.main.ScreenPointToRay (Input.mousePosition);
                   RaycastHit hit;
```

```
                                                    if (Physics.Raycast (ray, out hit,
maxRayDistance, collisionLayer)) {
                        //we're going to get the position from the contact point
                        m_HitTransform.position = hit.point;
                        Debug.Log (string.Format ("x:{0:0.######} y:{1:0.######}
z:{2:0.######}", m_HitTransform.position.x, m_HitTransform.position.y,
m_HitTransform.position.z));

                        m_HitTransform.rotation = hit.transform.rotation;
                    }
                }
                #else
                if (Input.touchCount > 0 && m_HitTransform != null)
                {
                    var touch = Input.GetTouch(0);
                    if (touch.phase == TouchPhase.Began || touch.phase ==
TouchPhase.Moved)
                    {
                        var screenPosition =
Camera.main.ScreenToViewportPoint(touch.position);
                        ARPoint point = new ARPoint {
                            x = screenPosition.x,
                            y = screenPosition.y
                        };

                      // prioritize reults types
                      ARHitTestResultType[] resultTypes = {

ARHitTestResultType.ARHitTestResultTypeExistingPlaneUsingExtent,
                      };

                      foreach (ARHitTestResultType resultType in resultTypes)
                      {
                          if (HitTestWithResultType (point, resultType))
                          {
                              return;
                          }
                      }
                    }
                }
                #endif

        }


    }
```